EXHIBIT 1

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

IMPLICIT NETWORKS, INC.,

Plaintiff,

v.

F5 NETWORKS, INC.,

Defendant.

_____/

IMPLICIT NETWORKS, INC.,

Plaintiff,

v.

HEWLETT-PACKARD COMPANY,

Defendant

_____/

IMPLICIT NETWORKS, INC.,

Plaintiff,

v.

JUNIPER NETWORKS, INC.,

Defendant.

_____/

Nos. C10-3365 SI; C 10-3746 SI; C 10-4234 SI

CLAIM CONSTRUCTION ORDER

On January 18 and 19, 2012, the Court held a *Markman* hearing regarding the construction of nine disputed claims in two patents owned by plaintiff. Having considered the arguments of counsel and the papers submitted, the Court construes the disputed claims as follows.

BACKGROUND

1. Procedural Background

Plaintiff filed Case No. 10-3365 against F5 Networks on July 30, 2010; Case No. 10-3746 against Hewlett-Packard Company on August 23, 2010; and Case No. 10-4234 against Juniper Networks, Inc., on September 20, 2010. In these related cases,[1] plaintiff accuses defendants' products of infringing two patents owned by plaintiff: U.S. Patent No. 6,629,163, as issued September 30, 2003 ("'163 Patent") and as it emerged after reexamination on June 22, 2010 ("'163 Reexam"); and U.S. Patent No. 7,711,857 ("'857 Patent"), issued May 4, 2010 as a continuation application from '163.[2]

2. Factual Background

According to the complaints in these cases, the heart of the patents' invention is a networking process where "discrete computer function[s], *e.g.*, processing http server requests over tcp/ip, streaming a video web-based client, or managing voice-over-ip calls, would be built into a discrete software module, called a 'bead.'" The system devised could "dynamically" "receive a stream of data – say video – determine what services were necessary to render that content and where the content was to be rendered, and then assemble – or string together – the requisite service beads (modules) at run-time." *See, e.g.*, FAC [Docket No. 31, Case No. 10-3365], ¶ 16. This system, according to plaintiff, dramatically departed from the prior art where a developer of applications had to anticipate who would use the applications and for which devices and content, and then build-in the ability to handle the anticipated demands. *Id.*, ¶ 11. The prior art model had many shortfalls, including an "ever-increasing complexity, cost, and processing overhead . . . Given that all anticipated uses had to be preconfigured at build-time, any unanticipated new use, *e.g.*, a different format or a different device, would simply break the system. The developer had to have the foresight to specify explicitly all possible configurations

---

[1] Other cases were also related, but those cases have been voluntarily dismissed. *See, e.g.*, Case Nos. 09-5628, 10-720, 10-3606.

[2] Although they are not being construed, the terms of U.S. Patent No. 7,730,211 (the "211 patent") are also involved, since the application for the '211 patent (U.S. Patent Application No. 11/933,093) was incorporated into the '857 patent and referred to in the '163 reexamination.

in advance, a difficult task in a rapidly changing world." *Id.*, ¶ 12.

As noted above, the '163 Patent entered reexamination in 2008 and emerged in June 2010 with additional limitations. All parties agree that the purpose and result of the reexamination was to distinguish the '163 series of patents from the prior art found in David Mosberger, "Scout: A Path-Based Operating System," Doctoral Dissertation Submitted to the University of Arizona. *See, e.g.*, Hosie Decl., [Docket No. 72], Ex. G at 11. In order to distinguish Mosberger, the '163 reexamination added a number of significant limitations, including "*dynamically* identifying a *non-predefined* sequence of components" for processing "messages", wherein "*dynamically identifying includes selecting individual components to create the non-predefined sequence of components*. . . ." '163 Reexam, Ex. D to Parties' Amended Joint Claim Construction and Prehearing Statement (emphasis in original showing terms added in reexam).

LEGAL STANDARD

Claim construction is a matter of law. *Markman v. Westview Instr., Inc.*, 517 U.S. 370, 372 (1996). Terms contained in claims are "generally given their ordinary and customary meaning." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005). "[T]he ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention." *Id.* at 1312. In determining the proper construction of a claim, a court begins with the intrinsic evidence of record, consisting of the claim language, the patent specification, and, if in evidence, the prosecution history. *Id.* at 1313; *see also Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996). "The appropriate starting point . . . is always with the language of the asserted claim itself." *Comark Communications, Inc. v. Harris Corp.*, 156 F.3d 1182, 1186 (Fed. Cir. 1998); *see also Abtox, Inc. v. Exitron Corp.*, 122 F.3d 1019, 1023 (Fed. Cir. 1997).

Accordingly, although claims speak to those skilled in the art, claim terms are construed in light of their ordinary and accustomed meaning, unless examination of the specification, prosecution history, and other claims indicates that the inventor intended otherwise. *See Electro Medical Systems, S.A. v. Cooper Life Sciences, Inc.*, 34 F.3d 1048, 1053 (Fed. Cir. 1994). The written description can provide

guidance as to the meaning of the claims, thereby dictating the manner in which the claims are to be construed, even if the guidance is not provided in explicit definitional format. *SciMed Life Systems, Inc. v. Advanced Cardiovascular Systems, Inc.*, 242 F.3d 1337, 1344 (Fed. Cir. 2001). In other words, the specification may define claim terms "by implication" such that the meaning may be "found in or ascertained by a reading of the patent documents." *Vitronics*, 90 F.3d at 1584 n.6.

In addition, the claims must be read in view of the specification. *Markman*, 52 F.3d at 978. Although claims are interpreted in light of the specification, this "does not mean that everything expressed in the specification must be read into all the claims." *Raytheon Co. v. Roper Corp.*, 724 F.2d 951, 957 (Fed. Cir. 1983). For instance, limitations from a preferred embodiment described in the specification generally should not be read into the claim language. *See Comark*, 156 F.3d at 1187. However, it is a fundamental rule that "claims must be construed so as to be consistent with the specification." *Phillips*, 415 F.3d at 1316. Therefore, if the specification reveals an intentional disclaimer or disavowal of claim scope, the claims must be read consistently with that limitation. *Id.*

Finally, the Court may consider the prosecution history of the patent, if in evidence. *Markman*, 52 F.3d at 980. The prosecution history limits the interpretation of claim terms so as to exclude any interpretation that was disclaimed during prosecution. *See Southwall Technologies, Inc. v. Cardinal IG Co.*, 54 F.3d 1570, 1576 (Fed. Cir. 1995). In most situations, analysis of this intrinsic evidence alone will resolve claim construction disputes. *See Vitronics*, 90 F.3d at 1583. Courts should not rely on extrinsic evidence in claim construction to contradict the meaning of claims discernable from examination of the claims, the written description, and the prosecution history. *See Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1308 (Fed. Cir. 1999) (citing *Vitronics*, 90 F.3d at 1583). However, it is entirely appropriate "for a court to consult trustworthy extrinsic evidence to ensure that the claim construction it is tending to from the patent file is not inconsistent with clearly expressed, plainly apposite, and widely held understandings in the pertinent technical field." *Id.* Extrinsic evidence "consists of all evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises." *Phillips*, 415 F.3d at 1317. All extrinsic evidence should be evaluated in light of the intrinsic evidence. *Id.* at 1319.

DISCUSSION

The parties' claim construction dispute centers on nine terms. The Court will address each in turn.[3]

1. Non-predefined sequence of components

Claim Term	Implicit's Proposed Construction	Defendants' Proposed Construction
"non-predefined sequence of components" '163 patent, Claims 1, 15, 35	Sequence of components changeable runtime. *Component*: plain meaning. In the alternative, one or more software routines.	A sequence of conversion routines that was not identified in or determinable from configuration information in place before the first packet of a message was received.

As an initial matter, the Court finds that **components** should be given the definition specifically provided for it in the specification: "software routines." *See* '163 Reexam (Reexamination Certificate) at Col. 1:27-29 & Col. 2:32 ("each component being a software routine").[4]

The more difficult question is the definition for **non-predefined**, a phrase that was added to the claims in the '163 reexamination. All parties agree that the phrase was introduced in order to distinguish '163 from the Mosberger prior art. *See, e.g.*, Plaintiff's Opening Claim Construction Brief [Case No. 10-3365 Docket No. 60] at 11; Defendants' Claim Construction Brief [Case 10-3365 Docket No. 63] at 8. Defendants attempt to impose a significant limitation on the term by arguing that the sequence of routines cannot be identified or "determinable from configuration information" in place before the first packet of a message is received. For support, defendants rely primarily on a September

[3] During the *Markman* hearing the parties agreed that one of the disputed terms **Identifying…a Sequence of Components…Such That the Output Format… Match[es] the Input Format of the Next Component** should be given its plain meaning. *See* Transcript pg. 73 [Docket No. 87 at 5]. While defense counsel clarified that the agreement to use "plain meaning" extended only to the "such that" part of the clause, *id*. at 73-74, defendants' claim construction demonstrative slides used during the second day of the *Markman* hearing confirm that defendants' "current proposed construction" for the whole phrase was its "plain meaning." Therefore, the Court finds that the term is not currently in dispute and will not construe it.

[4] The '163 and '857 patent specifications are largely identical. Therefore, the Court refers to the '163 Patent specification, unless otherwise specifically noted.

1, 2009 Amendment and Response submitted by Implicit during the '163 Reexam. *See* 9/1/09 Amendment and Response, Hogan Decl.[Docket No. 65], Ex. L. In that Response, Implicit cited to the '163 specification distinguishing prior art systems which "typically use predefined configuration information" to load the correct series of conversion routines that make up the "path." *Id*., at 18. However, Implicit's Amendment and Response makes clear that what it was disclaiming in the prior art was use of preconfigured sequences of routines, in other words preconfigured paths. *Id*., ("the sequence of conversion routines (or 'path') is not configured prior to receiving the first packet of a message."); *see also* 2/8/10 Amendment and Response, Hosie Decl., Ex. D at 16. Implicit did not disclaim the ability to create a sequence of conversion routines by relying in some part on predefined "configuration information," but only the use of pre-configured paths.[5]

Plaintiff's definition, however, is no more helpful as it attempts to introduce "changeable at runtime," words which themselves would need to be construed. The Court is mindful that "[t]he terms, as construed by the court, must 'ensure that the jury fully understands the court's claim construction rulings and what the patentee covered by the claims.'" *Power-One, Inc. v. Artesyn Techs., Inc*., 599 F.3d 1343, 1348 (Fed. Cir. 2010) (quoting *Sulzer Textil A.G. v. Picanol N.V.*, 358 F.3d 1356, 1366 (Fed. Cir. 2004)). Plaintiff's proposed definition does nothing to advance this goal. The proposed words do not find support in the specification and do not appear to be necessary because the claim itself identifies the time frame during which the sequence must be non-predefined – i.e., before "the first packet was received."

Non-predefined sequence of components, therefore, is construed as "a sequence of software routines that was not identified before the first packet of a message was received."

[5] To accept defendants' argument would also call into question a preferred embodiment of the '163 Patent. That embodiment uses the "label map get" feature, which plaintiff contends is a database that exists at the time of first packet inspection to "identify a sequence of conversion routines for processing the packet." '163 Specification at Col. 4:12-15.

2. Dynamically Identify[ing] a [Message Specific] Sequence of Components

Claim Term	Implicit's Proposed Construction	Defendants' Proposed Construction
"dynamically identify[ing] a [message specific] sequence of components" '857 patent, Claims 1, 4, 10 '163 patent, Claims 1, 15, 35	Selecting at runtime a sequence of components/selecting at runtime a sequence of components for the message	After receiving the first packet of a message, identifying and selecting individual components to create a sequence of conversion routines that was not identified in or determinable from predefined configuration information

The term **dynamically identify[ing]** was also added to the '163 Patent in the reexamination. The parties' proposed constructions are, essentially, restatements of their proposals for **non-predefined** discussed above. The Court also notes that **dynamically identify[ing]** is expressly defined in the claims themselves. For example, Claim 1 provides that "wherein **dynamically identifying** includes selecting individual components to create the non-predefined sequence of components after the first packet is received." '163 Reexam, Col. 1:36-39. Having already construed non-predefined as "a sequence of software routines that was not identified before the first packet of a message was received," and finding **dynamically identify[ing]** already defined in the claims, the Court rejects both parties' proposed constructions and adopts the plain meaning for **dynamically identify[ing] a [message specific] sequence of components**.

3. "Processing" [and Variants]

Claim Term	Implicit's Proposed Construction	Defendants' Proposed Construction
"…processing" and "all variants"[6] '163 patent Claims 1, 15, 35 '857 patent Claims 1, 4, 10	Manipulating data with a program.	Performing input to output format conversion.[7]

In proposing their preferred construction for **processing**, as well as **input/output format** discussed below, defendants are attempting to require that the patent cover only embodiments where the data in each of the packets of a message is "converted" in some manner.[8] The Court recognizes that the patent claims and specification repeatedly use the term "conversion." However, Claim 15 omits any reference to "conversion" and speaks only of "processing" packets of a message. Moreover, the '163 Specification itself explains, "a conversion routine may be used for routing a message, and may perform no conversion of the message." '163 Patent Col. 14:17-19. Therefore, defendants' argument that the claim language limits the invention to "conversion" of data within each packet is not well taken. With respect to the narrower issue of how to construe **processing**, the Court finds that plaintiff's proposal should be adopted and construes **processing [and variants]** as "manipulating data with a program."

[6] Variants include: "Processing [the] [packets of] [a/each] message," "Processing [the/a plurality of] packets of [a/the/each] message[s]," "Process[ing][es] the next packet of the message," etc.

[7] During the *Markman* hearing, defendants proposed this revised definition of **processing and variants**.

[8] *See, e.g.*, Defendants' Claim Construction Slides, "The Claim Language Limits the Invention to Conversion."

4. Input/Output Format

Claim Term	Implicit's Proposed Construction	Defendants' Proposed Construction[9]
"Input/Output Format." '163 patent Claim 1 '857 patent Claims 1, 4, 10	Input format: Structure or appearance of data to be processed. Output format: Structure or appearance of the data that results from processing.	Input format: Format for data that is input into a conversion routine. Output format: Format of data that it output from a conversion routine and is different from the input format.

Consistent with their arguments regarding **processing**, defendants attempt to limit **input/output format** to performing "conversion" of data so that the "format" of the output data of a packet is "different" from the input. Plaintiff attempts to maintain a broader functionality by relying on "manipulating" data and argues that defendants' attempted limitation is without support. As above, the Court agrees with plaintiff that "conversion" is not a necessary limitation with respect to the processing of each packet and finds that plaintiff's proposed construction is consistent with the claim language and the specification. *See, e.g.*, '163 Patent at Col. 6:31-33 ("[t]he terms 'media,' 'label,' and 'format' are used interchangeably to refer to the output of a protocol."). Therefore, **input/output format** are construed as the "structure or appearance of data to be processed" and the "structure or appearance of the data that results from processing."

[9] During the *Markman* hearing, defendants proposed this revised definition of **input/output format**.

5. Selecting Individual Components

Claim Term	Implicit's Proposed Construction	Defendants' Proposed Construction
"Selecting individual components" '163 patent Claims 1, 15, 35 '857 patent claims 1, 4, 10	Selecting components that are not bound together by a compiler.	Sequentially selecting the individual conversion routines of the sequence by comparing the input and output formats of the conversion routines.

The Court finds that neither party's proposed construction is particularly useful. Plaintiff's proposed construction includes words – "bound" and "compiler" – that themselves would need construing. However, defendants' proposal incorporates limitations that are not supported by the claim language or specification. For example, defendants point to no language in either the claims or the specification to support their intended limitation that the selection of individual components must be "sequential." Defendants contend that the prosecution history supports their "sequential" limitation by relying on plaintiff's October 23, 2009 interview summary with the PTO. 10/23/09 Summary, Hogan Decl., Ex. M at 2. There, plaintiff admitted that the "selecting individual components" limitations required "identifying the sequential order of the components based on the received packet." *Id.* However, identifying a sequential order is not the same as "sequentially selecting" individual components proposed by defendants.

Defendants also attempt to limit the means of **selecting individual components** to requiring a comparison of input and output formats of the software routines. Plaintiff argues that this "edge comparison" is only one method of selecting components covered by the claims. Plaintiff's Claim Construction Brief at 18-19. Plaintiff does not provide additional examples of methods for selecting individual components, other than referring to Label Map Get routine. *Id.* However, the Label Map Get routine itself ensures that the output format of each software routine "is compatible with the input format" of the next. *See* '163 Patent Col. 4:51-53. The Court also finds persuasive the repeated representations plaintiff made in the reexamination process regarding how the patent uses the format information of the packets to "identify individual components." In particular, "[t]he system of the '163

Patent uses this format information to dynamically identify components necessary for processing the entire message, such that the format of the output data of one module is compatible with the format of the input data of the next module." 9/1/09 Amendment and Response, Hogan Decl, Ex L at 22; *see also id.* ("Mosberger does not teach to define the input and output formats of the data that is processed by the modules, or to use these formats to make a run-time decision about how to assemble the modules."). The Court finds that, based on the claim language, teachings of the specification and the prosecution history, a necessary part of **selecting individual components** is determining the compatibility between the output of one software routine and the input of the next Therefore, **selecting individual components** is construed as "selecting the individual software routines of the sequence so that the input and output formats of the software routines are compatible."

6. "Create/Form [the…Sequence of Components]."

Claim Term	Implicit's Proposed Construction	Defendants' Proposed Construction
"Create/form [the…sequence of components]" **'163 patent Claims 1, 15, 35** **'857 patent Claims 1, 4, 10**	**Instantiate in memory.**	**Plain meaning.**

Plaintiff's proposal for this term attempts to incorporate words which themselves would need to be construed, *e.g.*, instantiate and memory. Plaintiff offers little support or reasoning for incorporating these additional words into the definition. Read in context, the Court does not find that the phrases **"create the sequence of components"** and "**form the sequence of components"** need any clarification. Therefore, c**reate/form [the . . . sequence of components]** is given its plain meaning.

7. Based on the First Packet of the Message

Claim Term	Implicit's Proposed Construction	Defendants' Proposed Construction
"based on the first packet of the message" **'163 patent Claim 15**	**Plain meaning, no construction needed. In the alternative, relying on information in the first packet of the message.**	**Based on the format of the data of the first packet.**

Plaintiff argues that defendants are attempting to introduce "format" to restrict the invention to relying on the underlying format of the data being processed – in order to identify the sequence of components – and exclude the ability to rely on header information. Defendants respond that the term "format" is used throughout the specifications and prosecution history as a key means by which the various components are placed into a sequence to process the packets of a message. While the input and output formats of the packets are a key aspect of selecting individual components to form the processing sequence, there is little to support limiting the demultiplexing method discussed in Claim 15 to using only information disclosed in or by the "format" of the data in first packet of a message. Therefore, the Court construes **based on the first packet of the message** as "relying on information in the first packet of the message."

8. Message[s]

Claim Term	Implicit's Proposed Construction	Defendants' Proposed Construction
"Message[s]" **'163 patent Claims 1, 15, 35** **'857 patent Claims 1, 4, 10**	**A collection or stream of data that is related in some way.**	**A collection of data in a particular format and that is related in some way, such as a stream of video or audio data or an email message.**

In the specification, plaintiff defined **message** as: "a collection of data that is related in some way, such as a stream of video or audio data or an email message." '163 Patent Col. 2:45-47. The question here is whether, in light of the examples given in that definition (*e.g.*, stream of video or email

message), the definition of **message** should be restricted to data in a particular format, as defendants contend. The Court finds it should not reach that question on claim constriction. **Message**, itself, has been clearly defined in the specification. Whether any particular form of data transmission falls within the express definition of **message** is a question for the trier of fact. Therefore, **message[s]** is construed as "a collection of data that is related in some way, such as a stream of video or audio data or an email message."

9. State Information

Claim Term	Implicit's Proposed Construction	Defendants' Proposed Construction
"State information" **'163 patent claims 1, 15, 35** **'857 patent Claims 1, 4, 10**	**Information specific to a component for a specific message.**	**Information specific to a conversion routine for a specific message that is not related to an overall path.**[10]

The parties agree that **state information** is information specific to a software routine (component) for a specific message. What the parties do not agree on is defendants' attempt to add, as a further limitation, "not related to an overall path." To support their additional limitation, defendants rely on Implicit's 9/1/09 Amendment and Response where Implicit – in distinguishing Mosberger – argued that "claim 1 is directed to a method in which state information for a specific component is stored on a component-by-component basis and is not information related to an overall path, as the Office Action describes Mosberger." 9/1/09 Amendment and Response, Hogan Decl., Ex, L at 24. Plaintiff, in Reply, asserts that the Amendment and Response was simply pointing out that the "novel element" the claim is directed to is the component-by-component rather than overall path storing, and that Implicit was not adding a limitation that the state information could not *also* relate to the overall sequence or path. Reply at 14. The Court, however, finds Implicit's statement in the reexamination was

[10] During the *Markman* hearing, defendants offered a new proposed construction.

clear. **State information** is "not information related to an overall path." This is consistent with the way state information is actually used in the claims and consistent with other language in the claims. *See, e.g.*, '163 Patent Col. 1:48-50 ("retrieving state information relating to performing the processing of the component with the previous packet of the message"); 1:54-56 ("storing state information relating to the processing of the component with packet for use when processing the next packed of the message"); *but see* Col. 1:39-42 ("storing an indication of each of the identified components so that the *non-predefined* sequence does not need to be re-identified for subsequent packets of the message"). **State information**, therefore, is construed as "information specific to a software routine for a specific message that is not information related to an overall path."

CONCLUSION

For the foregoing reasons and for good cause shown, the Court adopts the constructions set out above.

IT IS SO ORDERED.

Dated: February 29, 2012

SUSAN ILLSTON
United States District Judge

EXHIBIT 2

SPENCER HOSIE (CA Bar No. 101777)
shosie@hosielaw.com
GEORGE F. BISHOP (CA Bar No. 89205)
gbishop@hosielaw.com
DIANE S. RICE (CA Bar No. 118303)
drice@hosielaw.com
WILLIAM P. NELSON (CA Bar No. 196091)
wnelson@hosielaw.com
HOSIE RICE LLP
Transamerica Pyramid, 34th Floor
600 Montgomery Street
San Francisco, CA 94111
(415) 247-6000 Tel.
(415) 247-6001 Fax

Attorneys for Plaintiff
IMPLICIT NETWORKS, INC.

UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

IMPLICIT NETWORKS, INC.,

Plaintiff,

v.

JUNIPER NETWORKS, INC.,

Defendant.

Case No. C 10-4234 SI

PLAINTIFF'S DISCLOSURE OF ASSERTED CLAIMS AND INFRINGEMENT CONTENTIONS

In accordance with Rule 3-1 and Rule 3-2 of the Patent Local Rules of the United States District Court for the Northern District of California, Plaintiff IMPLICIT NETWORKS, INC. ("Plaintiff" or "Implicit") hereby provides its "Disclosure of Asserted Claims and Infringement Contentions" and "Document Production Accompanying Disclosure," as follows:

Disclosure Under Patent Local Rule 3-1(a)

Claims 1, 15, 26, 35, and 45 of U.S. Patent No. U.S. 6,629,163 C1 (the "'163 C1 Patent") and Claims 1, 4, and 10 of U.S. Patent No. U.S. 7,711,857 (the "'857 Patent") are infringed by Defendant JUNIPER NETWORKS, INC. ("Defendant" or "Juniper") pursuant to 35 U.S.C. § 271 (a-c, f-g).

Disclosure Under Patent Local Rule 3-1(b)

Each accused apparatus, product, device, process, method, act, or other instrumentality ("Accused Instrumentality") of Juniper – of which Plaintiff is currently aware – is identified, in Appendix A and incorporated by reference.

This disclosure is based on the present state of the Plaintiff's knowledge, without the benefit of any discovery from the Defendant or any other parties. The Plaintiff accordingly reserves the right to support its infringement action with additional allegations of infringement of other instrumentalities and of other claims, and with additional facts. The Plaintiff also reserves the right to modify the positions taken in these initial disclosures, based on later obtained materials, and/or based on information currently available, which the Plaintiff has not yet identified as significant.

Disclosure Under Patent Local Rule 3-1(c) (charts added as exhibits)

Exhibits A-B (Quality of Service functionality), C-D (Security functionality), E-F (Application Acceleration) identify specifically where each element of each asserted claim is found within each Accused Instrumentality.

This disclosure is based on the present state of the Plaintiff's knowledge, without the benefit of any discovery from the Defendant or any other parties. The Plaintiff

accordingly reserves the right to support its infringement action with additional allegations of infringement of other instrumentalities and of other claims, and with additional facts. The Plaintiff also reserves the right to modify the positions taken in these initial disclosures, based on later obtained materials, and/or based on information currently available which the Plaintiff has not yet identified as significant.

Disclosure Under Patent Local Rule 3-1(d)

Juniper has directly infringed each claim for which infringement is alleged herein. *See* Exhibits A-F hereto. Juniper directly infringes with respect to the products listed in Appendix A when it practices the infringing methods as described in Exhibits A-F, and when it makes, uses or sells a computer readable storage medium comprising the listed products with code for performing the infringing methods as described in Exhibits A-F.

Juniper's customers directly infringe when they use the products sold by Juniper that necessarily practice the patented method in its ordinary use as set forth in Exhibits A-F, or when they create a computer readable medium containing code for performing the patented methods by installing and configuring the products.

Juniper's acts of indirect infringement include actively inducing infringement, and selling the products listed above knowing that they are especially made for use in an infringement, and not a staple article or commodity of commerce suitable for substantial non-infringing use. Juniper knowingly and actively induces, aids, and abets its customer's infringement. The acts of Juniper inducing or contributing to direct infringement by others include the following:

Juniper sells, markets and advertises its products listed in Appendix A, knowing that customers will use them to practice the patented methods, *e.g.*, performing packet inspection on the first packets of a message to dynamically invoke a sequence of components to process the message, and storing state information so that subsequent packets of the message are processed accordingly, and knowing that customers will

install and configure the software and hardware, thereby creating a computer-readable medium containing instructions for performing those methods. In addition, Juniper provides inducing services that include design, development, training and support, that solicit, instruct, train and support its customers to practice the patented methods and to create a computer readable medium for practicing the patented methods by installing and configuring the software and hardware. Thus, for example, when Juniper sells its routers, switches and gateway products, it will market and advertise them, and after the sale, provide the customer with extensive support, providing knowledge, tools, libraries and sample code to its program developers in order to build, deploy and maintain network architectures that practice the patented methods of the '163 and '857 patents, and to create the computer readable medium of the claims by installing and configuring the accused products.

Defendant Juniper's distribution or sale of its products identified in Appendix A induce its customers and contributes to their infringement.

Juniper's acts of direct infringement, and its customers' direct infringement, occur in industries and with customers including those set forth on Juniper's website, which are known to Juniper.

Disclosure Under Patent Local Rule 3-1(e)

Each element of each claim as set forth in Exhibits A-F is literally present or, in the alternative, is present under the doctrine of equivalents in the Accused Instrumentalities.

Disclosure Under Patent Local Rule 3-1(f)

The '163 C1 Patent is based on Application No. 09/474,664 (filed December 29, 1999), and as a result, the asserted claims of the '163 C1 Patent claim December 29, 1999, as their priority date.

The '857 Patent is based on Application No. 11/933,022 (filed October 31, 2007), which is a continuation of Application No. 10/636,314 (filed August 26, 2003), which is a continuation of Application No. 09/474,664, filed on December 29, 1999,

now Patent No. 6,629,163 C1, and as a result, the asserted claims of the '857 Patent claim December 29, 1999, as their priority date.

Disclosure Under Patent Local Rule 3-1(g)

For the purpose of preserving the right to rely, for any purpose, on the assertion that its own apparatus, product, device, process, method, act, or other instrumentality practices the claimed invention, the Plaintiff identifies the following product(s):

Portal
Strings
The following Strings packages:
Strings Core
Namespace
Package Manager
Network Support
HTTP
Strings Discovery
RADkit Support
Strings Network
Host Network
Synchronization
System Status
NAT
Media Routing
Bridge
IP Route
HTTP Director
Open GL
POP3 Client
SMTP Client
Mini Browser
VoIP
PBX Gateway
Streaming Media Storage
TV Tuner
Audio Pack
Video Pack
Time Shift
Text to Speech
Direct Show Gateway
Real Audio Gateway
Windows Media Gateway
Fax

UPnP
IPv6
Mozilla
EPG
Web Services
Encryption
Authentication
DRM
Remote Win32 Client
Remote Win32 Server
Speech To Text

Disclosure Under Patent Local Rule 3-1(h)

The Plaintiff claims willful infringement on the part of the Defendant at this time, as Juniper is continuing to use, sell and import the accused product line despite the filing of this action. The Plaintiff reserves the right to modify the positions taken in these Initial Disclosures, based on later obtained materials and/or based on information currently available that the Plaintiff has not yet identified as significant.

Document Production Under Patent Local Rule 3-2

The Plaintiff objects to the requirements of this production to the extent that it calls for documents protected by the attorney-client privilege. Further, in producing these documents, the Plaintiff does not admit or concede the relevancy, materiality, authenticity, or admissibility as evidence of any of these documents. All objections to the use, at trial or otherwise, of any document produced are hereby expressly reserved. The Plaintiff's discovery and investigation in connection with this lawsuit is commencing and will continue throughout. As a result, the Plaintiff produces these documents without prejudice as to the right to produce additional documents after considering documents obtained or reviewed through further discovery or investigation. Subject to and without waiving its objections, the Plaintiff produces responsive documents as follows:

Patent L.R. 3-2(a): None to produce;

Patent L.R. 3-2(b): The inventor's notebook is being produced subject to the

protective order at IMP00001 - 00250;

Patent L.R. 3-2(c): Plaintiff has produced the file histories of the '163 C1 and '857 Patents at IMP089974 - 090288 and IMP089788 - 089973, respectively;

Patent L.R. 3-2(d): Plaintiff has produced assignment documents associated with the '163 and '857 Patents at IMP089586 and IMP089883, respectively; and

Patent L.R. 3-2(e): None to produce.

Undersigned counsel hereby certifies that to the best of his knowledge, information, and belief, formed after an inquiry that is reasonable under the circumstances, the information contained in this Disclosure and the attached Exhibits and the production of documents pursuant to Patent L.R. 3-2 is complete and correct at the time of certification.

Dated: May 23, 2011

Respectfully submitted,

/s/ Spencer Hosie

SPENCER HOSIE (CA Bar No. 101777)
shosie@hosielaw.com
GEORGE F. BISHOP (CA Bar No. 89205)
gbishop@hosielaw.com
DIANE S. RICE (CA Bar No. 118303)
drice@hosielaw.com
WILLIAM P. NELSON (CA Bar No. 196091)
wnelson@hosielaw.com
HOSIE RICE LLP
Transamerica Pyramid, 34th Floor
600 Montgomery Street
San Francisco, CA 94111
(415) 247-6000 Tel.
(415) 247-6001 Fax

Attorneys for Plaintiff
IMPLICIT NETWORKS, INC.

APPENDIX A

(Juniper Products Containing Infringing Technologies)

Application Acceleration Category

1. DX3200 Series Application Acceleration Platform (deprecated)
2. DX3250 Series Application Acceleration Platform (deprecated)
3. DX3280 Series Application Acceleration Platform (deprecated)
4. DX3600 Series Application Acceleration Platform (deprecated)
5. DX3650 / DX3650 FIPS Application Acceleration Platform (deprecated)
6. DX3670 Application Acceleration Platform (deprecated)
7. DX3680 Application Acceleration Platform (deprecated)
8. WX Stack Series Data Center Acceleration (deprecated)
9. WX 15 Series Application Acceleration Platform (deprecated)
10. WX 20 Series Application Acceleration Platform (deprecated)
11. WX 50 Series Application Acceleration Platform
12. WX 60 Series Application Acceleration Platform (deprecated)
13. WX 80 Series Application Acceleration Platform
14. WX 100 Series Application Acceleration Platform (deprecated)
15. WXC 250 Series Application Acceleration Platform (deprecated)
16. WXC 500 Series Application Acceleration Platform (deprecated)
17. WXC 590 Series Application Acceleration Platform
18. WXC 1800 Series Application Acceleration Platform
19. WXC 2600 Series Application Acceleration Platform
20. WXC 3400 Series Application Acceleration Platform
21. J2320 Series Router with ISM WXC 200 installed
22. J2350 Series Router with ISM WXC 200 installed
23. J4350 Series Router with ISM WXC 200 installed
24. J6350 Series Router with ISM WXC 200 installed
25. Junos Pulse

QOS Category

1. EX2200 Series Switches
2. EX2500 Series Switches
3. EX3200 Series Switches
4. EX4200 Series Switches
5. EX4500 Series Switches
6. EX8208 Series Switches
7. EX8216 Series Switches
8. QFX3500 Series Switches
9. CTP150 Series Circuit to Packet Platform
10. CTP1002 Series Circuit to Packet Platform
11. CTP1004 Series Circuit to Packet Platform
12. CTP1012 Series Circuit to Packet Platform
13. CTP2008 Series Circuit to Packet Platform
14. CTP2024 Series Circuit to Packet Platform

15. CTP2056 Series Circuit to Packet Platform
16. E120 Series Broadband Services Router
17. E320 Series Broadband Services Router
18. ERX310 Series Broadband Services Router
19. ERX705 Series Broadband Services Router
20. ERX710 Series Broadband Services Router
21. ERX1410 Series Broadband Services Router
22. ERX1440 Series Broadband Services Router
23. J2300 Series Router (deprecated)
24. J2320 Series Router
25. J2350 Series Router
26. J4300 Series Router (deprecated)
27. J4350 Series Router
28. J6300 Series Router (deprecated)
29. J6350 Series Router
30. LN1000 Series Mobile Secure Router
31. M5 Series Router (deprecated)
32. M7i Series Router
33. M10 Series Router (deprecated)
34. M10i
35. M20 Series Router (deprecated)
36. M40 Series Router (deprecated)
37. M40e Series Router
38. M120 Series Router
39. M160 Series Router (deprecated)
40. M320 Series Router
41. MX5 Series Router
42. MX10 Series Router
43. MX40 Series Router
44. MX80 Series Router
45. MX240 Series Router
46. MX480 Series Router
47. MX960 Series Router
48. T320 Series Router
49. T640 Series Router
50. T1600 Series Router
51. T4000 Series Router
52. TX Matrix Series Router
53. TX Matrix Plus Series Router

Security Category

1. J2320 Series Router
2. J2350 Series Router
3. J4350 Series Router
4. J6350 Series Router
5. LN1000 Series Mobile Secure Router

6. NetScreen-5200 Series
7. NetScreen-5400 Series
8. MX240 Series Router with Multiservices DPC installed
9. MX480 Series Router with Multiservices DPC installed
10. MX960 Series Router with Multiservices DPC installed
11. M7i Series Router with Multiservices PIC installed
12. M10i Series Router with Multiservices PIC installed
13. M40e Series Router with Multiservices PIC installed
14. M120 Series Router with Multiservices PIC installed
15. M320 Series Router with Multiservices PIC installed
16. T320 Series Router with Multiservices PIC installed
17. T640 Series Router with Multiservices PIC installed
18. T1600 Series Router with Multiservices PIC installed
19. TX Matrix Series Router with Multiservices PIC installed
20. IDP 10 Series Intrusion Detection and Prevention Appliance (deprecated)
21. IDP 50 Series Intrusion Detection and Prevention Appliance (deprecated)
22. IDP 75 Series Intrusion Detection and Prevention Appliance
23. IDP 100 Series Intrusion Detection and Prevention Appliance (deprecated)
24. IDP 200 Series Intrusion Detection and Prevention Appliance (deprecated)
25. IDP 250 Series Intrusion Detection and Prevention Appliance
26. IDP 500 Series Intrusion Detection and Prevention Appliance (deprecated)
27. IDP 600 C/600 F Series Intrusion Detection and Prevention Appliance (deprecated)
28. IDP 800 Series Intrusion Detection and Prevention Appliance
29. IDP 1000 Series Intrusion Detection and Prevention Appliance (deprecated)
30. IDP 1100C 1100F Series Intrusion Detection and Prevention Appliance (deprecated)
31. IDP 4500 Series Intrusion Detection and Prevention Appliance (deprecated)
32. IDP 6500 Series Intrusion Detection and Prevention Appliance (deprecated)
33. IDP 8200 Series Intrusion Detection and Prevention Appliance
34. ISG1000 Series Integrated Security Gateway with Optional IPS
35. ISG2000 Series Integrated Security Gateway with Optional IPS
36. SRX100 Series Services Gateway
37. SRX210 Series Services Gateway
38. SRX220 Series Services Gateway
39. SRX240 Series Services Gateway
40. SRX650 Series Services Gateway
41. SRX1400 Series Services Gateway
42. SRX3400 Series Services Gateway
43. SRX3600 Series Services Gateway
44. SRX5600 Series Services Gateway
45. SRX5800 Series Services Gateway

CERTIFICATE OF SERVICE

I, Jerry Shaw, am a citizen of the United States and am employed in the County of San Francisco, State of California. I am over the age of 18 years and am not a party to the within action. My business address is Hosie Rice LLP, Transamerica Pyramid, 34th Floor, 600 Montgomery Street, San Francisco, California, 94111.

On May 23, 2011, I served the following attached

PLAINTIFF'S DISCLOSURE OF ASSERTED CLAIMS AND INFRINGEMENT CONTENTIONS

via Federal Express at San Francisco, California, addressed to the following parties:

DAVID C. MCPHIE
dmcphie@irell.com
REBECCA L. CLIFFORD
rclifford@irell.com
Irell & Manella LLP
840 Newport Center Drive, Suite 400
Newport Beach, CA 92660-6324

MORGAN CHU
mchu@irell.com
JONATHAN S. KAGAN
jkagan@irell.com
IRELL & MANELLA LLP
1800 Avenue of the Stars, Suite 900
Los Angeles, CA 90067-4276

Attorneys for Defendant
Juniper Networks, Inc.

I certify under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

DATED: May 23, 2011

/s/ Jerry Shaw
Jerry Shaw

EXHIBIT 3

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

'163 C1 Patent: Claim 1	**Method and System for Demultiplexing a First Sequence of a Packet, Components to Identify Specific Components, Wherein Subsequent Components Are Processed Without Re-Identifying Components**
1. Preamble. A method in a computer system for processing a message having a sequence of packets,	Juniper Networks, Inc. provides networking products, in the form of equipment and software, specializing in the field of Ethernet and IP networking. Juniper supplies a number of different types of products for constructing these networks, depending on the location or function in the network. For security functions in the network, the accused Juniper products utilize Intrusion Detection and Prevention (IDP), and Unified Threat Management (UTM). IDP functions are also known as "Deep Inspection", or as "Intrusion Detection System" (IDS), or as "Intrusion Prevention System" (IPS). UTM functions are also known as Network Anti-Virus, Network Anti-Spam, Web Filtering, or Content Filtering. The accused products (IDP and UTM functionality) are present in appliances, switches, routers, and modules of the SRX Series, J Series, NetScreen Series, ISG Series, SSG Series, and IDP Series. The accused products have these features implemented as part of/in conjunction with the embedded Operating System which is included in Junos OS. **Evidence '163 C1 Pre(1)**

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

JUNIPER NETWORKS

DATASHEET

SECURITY PRODUCTS COMPARISON MATRIX

FIREWALL/VPN PRODUCTS	INTERFACES	MAX THROUGHPUT	MAX SESSIONS	MAX POLICIES	VIRTUAL SYSTEMS	VIRTUAL LANS	SECURITY ZONES	VIRTUAL ROUTERS	HIGH AVAILABILITY[1]	ROUTING	DEEP INSPECTION/ IPS	INTEGRATED ANTIVIRUS/ ANTISPAMS	WEB FILTERING (INTEGRATED/ EXTERNAL)
SRX5800	40 SFP GigE, 4 XFP 10Gig (SR or LR), 16 GigE (TX or XFP) FlexIOC, or 4 XFP 10Gig (SR or LR) FlexIOC	120 Gbps firewall, 30 Gbps 3DES/AES VPN, 30 Gbps IPS	14,000,000	80,000	Future release	4,096	512	500	A/P, A/A	OSPF, BGP, RIPv1/v2, Multicast	Yes / Yes	No	No / Yes
SRX5600	40 SFP GigE, 4 XFP 10Gig (SR or LR), 16 GigE (TX or XFP) FlexIOC, or 4 XFP 10Gig (SR or LR) FlexIOC	60 Gbps firewall, 15 Gbps 3DES/AES VPN, 15 Gbps IPS	9,000,000	80,000	Future release	4,096	256	500	A/P, A/A	OSPF, BGP, RIPv1/v2, Multicast	Yes / Yes	No	No / Yes
SRX3600	8 10/100/1000 + 4 SFP (on-board) 16 SFP GigE, 16 10/100/1000, or 2 XFP 10Gig (SR or LR)	30 Gbps firewall, 10 Gbps 3DES/AES VPN, 10 Gbps IPS	6,000,000	40,000	Future release	4,096	256	500	A/P, A/A	OSPF, BGP, RIPv1/v2, Multicast	Yes / Yes	No	No / Yes
SRX3400	8 10/100/1000 + 4 SFP (on-board) 16 SFP GigE, 16 10/100/1000, or 2 XFP 10Gig (SR or LR)	20 Gbps firewall, 6 Gbps 3DES/AES VPN, 6 Gbps IPS	3,000,000	40,000	Future release	4,096	256	500	A/P, A/A	OSPF, BGP, RIPv1/v2, Multicast	Yes / Yes	No	No / Yes
SRX1400	6 10/100/1000 + 6 SFP or 6 10/100/1000 + 3 SFP and 3 10GbE (on board) 16 SFP GbE, 16 10/100/1000, or 2 XFP 10GbE	10 Gbps firewall, 2 Gbps firewall and IPS, 2 Gbps 3DES/AES VPN	512,000	40,000	Future release	4,096	256	500	A/P, A/A*	OSPF, BGP, RIPv1/v2, Multicast	Yes / Yes	No	No / Yes
SRX650	4 10/100/1000, 8 I/O slots supporting GE, PoE, SFP, T1, E1	7 Gbps firewall, 1.5 Gbps 3DES/AES VPN, 900 Mbps IPS	512,000	8,192	N/A	4,096	128	60	A/P, A/A	OSPF, BGP, RIPv1/ v2, MPLS, Multicast	No / Yes	Yes	Yes
SRX240	16 10/100/1000, optional PoE, 4 I/O slots supporting SFP, ADSL, ADSL2, ADSL2+, Serial, T1, E1	1.5 Gbps firewall, 250 Mbps 3DES/AES VPN, 250 Mbps IPS	64,000/ 128,000[4]	4,096	N/A	512	32	20	A/P, A/A	OSPF, BGP, RIPv1/ v2, MPLS, Multicast	No / Yes	Yes	Yes
SRX210	2 10/100/1000 + 6 10/100, optional PoE, 1 I/O slot supporting SFP, ADSL, ADSL2, ADSL2+, Serial, T1, E1	750 Mbps firewall, 75 Mbps 3DES/AES VPN, 80 Mbps IPS	32,000/ 64,000[4]	512	N/A	64	12	10	A/P, A/A	OSPF, BGP, RIPv1/ v2, MPLS, Multicast	No / Yes	Yes	Yes
SRX100	8 10/100	650 Mbps firewall, 65 Mbps 3DES/AES VPN, future IPS[4]	16,000/ 32,000[4]	384	N/A	16	10	3	A/P, A/A	OSPF, BGP, RIPv1/ v2, MPLS, Multicast	No / Yes[4]	Yes / Yes[4]	Yes
J6350	4 10/100/1000 and 6 I/O slots, supporting SFP, Serial, T1, E1, DS3, E3, ADSL, ADSL2, ADSL2+, G.SHDSL, 10/100/1000	2 Gbps firewall, 1 Gbps 3DES/AES VPN	256,000	10,384	N/A	1024	50	30	A/P, A/A	OSPF, BGP, RIPv1/v2	No / Yes	Yes	Yes
J4350	4 10/100/1000 and 6 I/O slots, supporting SFP, Serial, T1, E1, DS3, E3, ADSL, ADSL2, ADSL2+, G.SHDSL, 10/100/1000	1.6 Gbps firewall, 600 Mbps 3DES/AES VPN	128,000	5,192	N/A	512	50	30	A/P, A/A	OSPF, BGP, RIPv1/v2	No / Yes	Yes	Yes
J2350/J2320	4 10/100/1000 and 5 I/O slots (3 in J2320) supporting Serial, ISDN BRI S/T, T1, E1, ADSL, ADSL2, ADSL2+, G.SHDSL	750 Mbps firewall, (600 Mbps w/ J2320), 160 Mbps 3DES/AES VPN (140 Mbps w/ J2320)	128,000	2,048	N/A	256	50	25/20	A/P, A/A	OSPF, BGP, RIPv1/v2	No / Yes	Yes	Yes
NetScreen-5400/ NetScreen-5200[3]	8 mini-GBIC (SX, LX or TX), or 2 XFP 10Gig (SR or LR)	30/10 Gbps firewall, 15/5 Gbps 3DES/AES VPN	2,000,000/ 1,000,000	40,000	Up to 500	4,094	16 + up to 1,000 additional[2]	3 + up to 500 additional[2]	A/P, A/A, F/M	OSPF, BGP, RIPv1/v2	Yes / No	No	No / Yes
ISG2000 w/ optional IPS	Up to 16 mini-GBIC (SX, LX, or TX), up to 8 10/100/1000, up to 28 10/100, up to 4 XFP 10Gig (SR or LR)	4 Gbps firewall, 2 Gbps 3DES/AES VPN, 2 Gbps IPS	1,000,000[4]	30,000	Up to 250	4,094[4]	26 + up to 500 additional[2]	3 + up to 250 additional[2]	A/P, A/A, F/M	OSPF, BGP, RIPv1/v2	Yes / Yes	No	Yes / Yes
ISG1000 w/ optional IPS	Up to 16 mini-GBIC (SX, LX, or TX), up to 8 10/100/1000, up to 28 10/100, up to 4 XFP 10Gig (SR or LR)	2 Gbps firewall, 1 Gbps 3DES/AES VPN, 1 Gbps IPS	500,000[4]	10,000	Up to 50	4,094[4]	26 + up to 500 additional[2]	3 + up to 250 additional[2]	A/P, A/A, F/M	OSPF, BGP, RIPv1/v2	Yes / Yes	No	Yes / Yes
SSG550M/ SSG520M	4 10/100/1000 and 6 I/O slots supporting SFP, Serial, T1, E1, DS3, E3, ADSL, and ADSL2 (SSG550M only), ADSL2+, G.SHDSL, 10/100/1000	1+ Gbps firewall, (650+ Mbps w/ SSG520M), 300 Mbps 3DES/AES VPN (300 Mbps w/ SSG520M)	256,000/ 128,000	4,000	N/A	150/125	60	16 /11	A/P, A/A	OSPF, BGP, RIPv1/v2	Yes / No	Yes	Yes
SSG350M/ SSG320M	4 10/100/1000 and 5 I/O slots (3 in SSG320M) supporting Serial, ISDN BRI S/T (SSG350M only), T1, E1, ADSL, ADSL2, ADSL2+, G.SHDSL	550+ Mbps firewall (450+ Mbps w/ SSG320M), 225 Mbps 3DES/AES VPN (175 Mbps w/ SSG320M)	128,000/ 64,000	2,000	N/A	125	40	8/5	A/P, A/A	OSPF, BGP, RIPv1/v2	Yes / No	Yes	Yes
SSG140	8 10/100 + 2 10/100/1000 + 4 I/O slots supporting T1, E1, ISDN BRI S/T, Serial, ADSL2+, G.SHDSL, 10/100/1000, SFP	350+ Mbps firewall, 100 Mbps 3DES/AES VPN	48,000	1,000	N/A	100	40	6	A/P, A/A	OSPF, BGP, RIPv1/v2	Yes / No	Yes	Yes
SSG20 SSG20 Wireless	5 10/100 + 2 I/O slots supporting T1, E1, V.92, ISDN BRI S/T, SFP, Serial, or ADSL2+, optional 802.11a/b/g	160 Mbps firewall, 40 Mbps 3DES/AES VPN	8,000/ 16,000[4]	200	N/A	10/50[4]	8	3/4	A/P[4], A/A, dial backup	OSPF, BGP, RIPv1/v2	Yes / No	Yes	Yes
SSG5 SSG5 Wireless	7 10/100 with factory configured V.92 or ISDN BRI S/T or RS232 Serial/AUX, optional 802.11a/b/g	160 Mbps firewall, 40 Mbps 3DES/AES VPN	8,000/ 16,000[4]	200	N/A	10/50[4]	8	3/4	A/P[4], A/A, dial backup	OSPF, BGP, RIPv1/v2	Yes / No	Yes	Yes

IDP SERIES INTRUSION DETECTION AND PREVENTION APPLIANCES	MAX THROUGHPUT	MAX SESSIONS	OPERATIONAL MODES	DETECTION MECHANISMS	SIGNATURE UPDATES	INTERFACES	HIGH AVAILABILITY
IDP8200	10 Gbps	5,000,000	Passive sniffer Inline bridge Inline Proxy-ARP Inline router	8 including Stateful Signatures, Protocol Anomalies and Backdoor Detection	Daily and emergency	Configurable up to 16 CG or 16 Fiber SX/BYP or 8 10 G fiber traffic, 1 CG mgmt and 1 CG HA ports	Optional integrated bypass for copper and fiber for all traffic ports
IDP800	1 Gbps	1,000,000				10 CG traffic, 1 CG mgmt and 1 CG HA ports	
IDP250	300 Mbps	300,000				8 CG traffic, 1 CG mgmt and 1 CG HA ports	Integrated bypass
IDP75	150 Mbps	100,000				2 CG traffic + 1 CG mgmt ports	

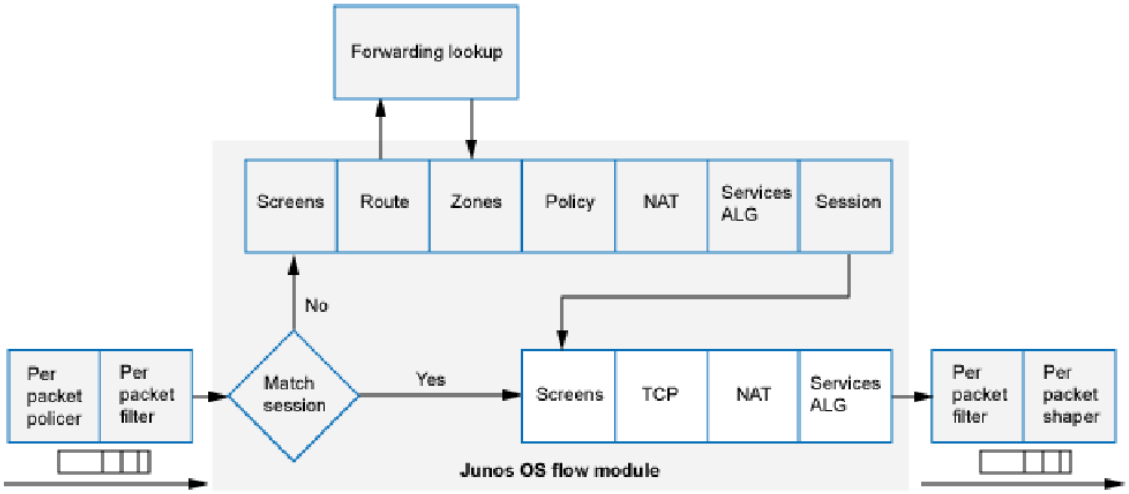
Source: *Security Products Comparison Matrix*, Published by Juniper Networks, Inc., November 2010, http://www.juniper.net/us/en/local/pdf/datasheets/1000265-en.pdf

Evidence '163 C1 Pre(2)

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Junos OS for SRX Series Services Gateways integrates the world-class network security and routing capabilities of Juniper Networks. Junos OS includes a wide range of packet-based filtering, class-of-service (CoS) classifiers, and traffic-shaping features as well as a rich, extensive set of flow-based security features including policies, screens, network address translation (NAT), and other flow-based services. Traffic that enters and exits services gateway is processed according to features you configure, such as packet filters, security policies, and screens. For example, the software can determine: • Whether the packet is allowed into the device • Which firewall screens to apply to the packet • The route the packet takes to reach its destination • Which CoS to apply to the packet, if any • Whether to apply NAT to translate the packet's IP address • Whether the packet requires an Application Layer Gateway (ALG)

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case



Flow-based packet processing treats related packets, or a stream of packets, in the same way. Packet treatment depends on characteristics that were established for the first packet of the packet stream, which is referred to as a flow.

Packet-based, or stateless, packet processing treats packets discretely. Each packet is assessed individually for treatment.

A packet undergoes flow-based processing after packet-based filters and some screens have been applied to it. All flow-based processing for a single flow occurs on a single System Processing Unit (SPU). An SPU processes the packets of a flow according to the security features and other services configured for the session.

A flow is a stream of related packets that meet the same matching criteria and share the same characteristics. Junos OS treats packets belonging to the same flow in the same manner.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

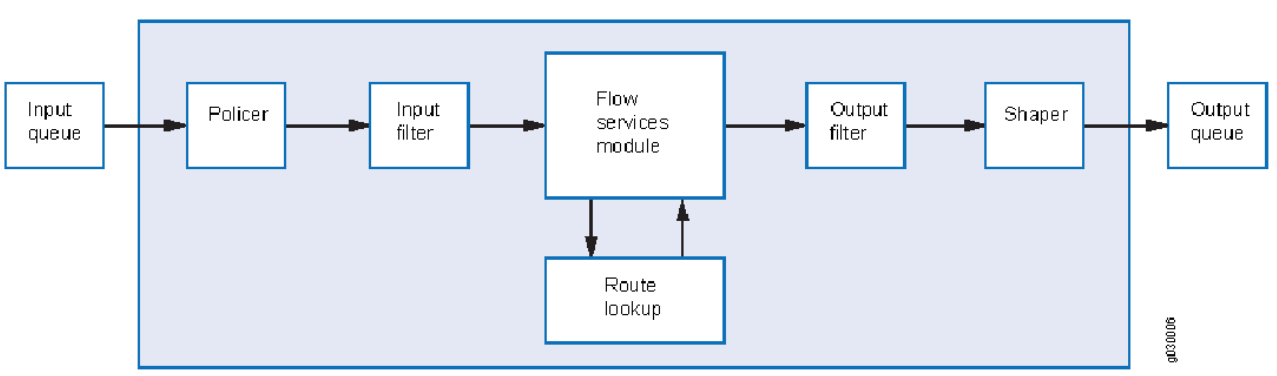
Source: *Junos OS Security Configuration Guide*, Published by Juniper Networks, Inc., March 2011, pages 4-5, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

Evidence '163 C1 Pre(3)

Junos OS for J Series Services Routers integrates the world-class network security and routing capabilities of Juniper Networks Operating System.

Traffic that enters and exits a services router running Junos OS is processed according to features you configure, such as security policies, packet filters, and screens. For example, the software can determine:

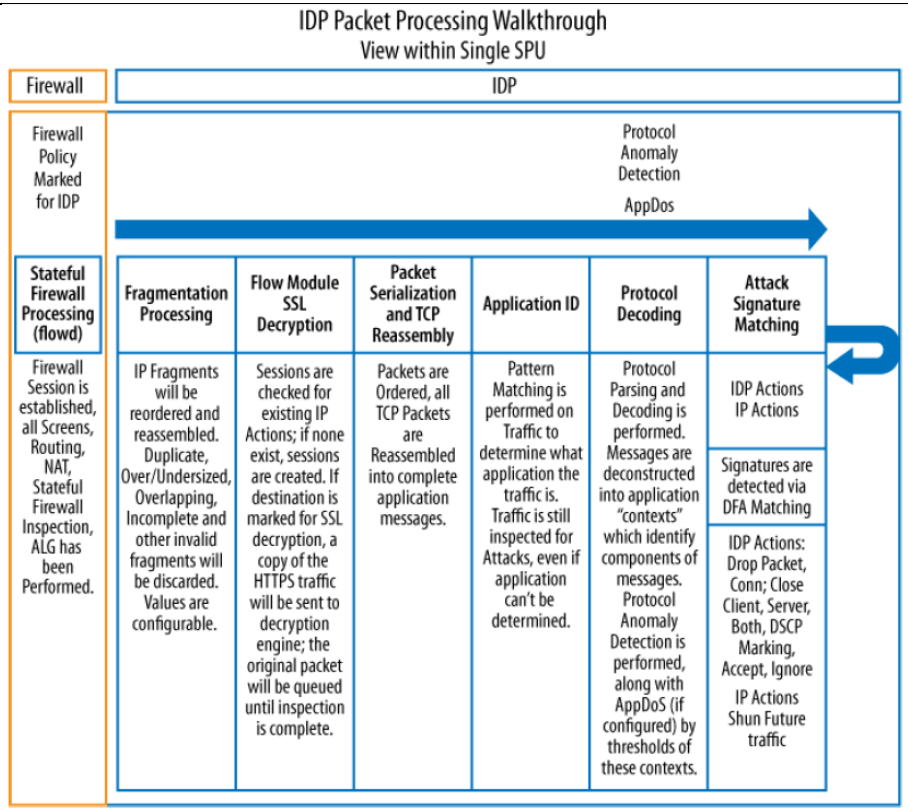
- Whether the packet is allowed into the router
- Which class of service (CoS) to apply to the packet, if any
- Which firewall screens to apply to the packet
- Whether to send the packet through an IPsec tunnel
- Whether the packet requires an Application Layer Gateway (ALG)
- Whether to apply Network Address Translation (NAT) to translate the packet's address
- Which route the packet uses to reach its destination



Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Packets that enter and exit a services router running Junos OS undergo both packet-based and flow-based processing. A device always processes packets discretely. Packet treatment depends on characteristics that were established for the first packet of the packet stream. A packet undergoes flow-based processing after any packet-based filters and policers have been applied to it. A flow is defined as a set of packets coming from the same source/destination addresses, source/destination ports (when applicable), protocol, and ingress/egress zones. Flows are time bound so it is possible to have packets that, while fitting the previous definition belong to different flows. For example, when an existing session is initiated and terminated, after which a new session is established using the exact same parameters as the previous session, the packets would belong to different flows. **Source:** *Junos OS Security Configuration Guide*, Published by Juniper Networks, Inc., March 2011, pages 94, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf
1a. the method comprising: providing a plurality of components, each component being a software routine for converting data with an input format into data with an output format;	The accused products provide components which operate on the data in sequence, the output of one component being the input of the next; they also perform IPS algorithm processing. **Evidence '163 C1 1a(1)**

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case



Within the IPS engine there are several stages of processing, as illustrated above. IPS processing on the SRX can be broken down into eight general stages of processing:

Stage 1: Fragmentation processing

The first thing that must happen before you can really get to the inspection is that the SRX must process fragmented traffic (if present). To ensure that common IDS evasion techniques using fragmentation are not

	effective, it rebuilds any fragmented traffic from a Layer 3 perspective. This stage also provides countermeasures against fragment-based attacks such as missing fragments, underlapping or overlapping fragments, duplicate fragments, and other fragment-based anomalies. Many of these values are also configurable in the IPS sensor configuration section, although defaults should suffice in most cases. Stage 2: IPS flow setup After any Layer 3 fragments are processed, the SRX examines the traffic to see whether it has an existing session for it or if there is an existing session which might need some special processing. The IPS session table is different from the firewall session table, because additional IPS state related to the traffic is required. Stage 3: SSL decryption (if applicable) If SSL decryption is configured, and traffic is destined to a web server that is configured to be decrypted, decryption happens in this phase. Stage 4: Serialization and reassembly For accurate IPS processing, all messages must be processed in order, in a flow, and the messages must be reassembled if they span multiple packets. Without reassembly, an IPS engine can be easily evaded, which would result in lots of false positives. The SRX IPS engine ensures that before traffic is processed, it is ordered and reassembled in this stage of the processing. Stage 5: Application identification The SRX has the ability to detect what application is running on any Layer 4 port. This is important because it allows the device to determine what traffic is running in a given flow regardless of whether it is running on a standard port. Even if the application cannot be identified, the SRX can still inspect it as a bytestream. This stage typically happens within the first couple of kilobytes of traffic, and the SRX utilizes both directions of the traffic to identify the application. Stage 6: Protocol decoding

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Once the application is identified (or is simply classified as a stream), the SRX decodes the application from a protocol level, a process known as protocol decoding. Protocol decoding allows the SRX to chop up the traffic into contexts, which are specific parts of different messages. Contexts are very important to IPS processing because they allow the SRX to look for attacks in the specific location where they actually occur, not just blindly by byte matching across all traffic that passes through the SRX. After all, you wouldn't want the SRX to block an email conversation between you and a peer discussing the latest exploit; you would only want the SRX to block the exploit in the precise location where it actually occurs. At the time of this writing, the SRX supports almost 600 application contexts. Contexts are one of the ways that the SRX seeks to eliminate false positives. The protocol decoding stage is also where the SRX performs protocol anomaly protection and Application Distributed Denial of Service (AppDDoS) protection, both of which we will discuss later in this chapter. Stage 7: Stateful signature detection The attack objects that rely on signatures (rather than anomaly detection) are processed in the stateful signature stage of the device's processing. These signatures are not blind pattern matches, but are highly accurate stateful signatures that not only match attacks within the contexts in which they occur, but also can be composed of multiple match criteria (using Boolean expressions between individual criteria). Typically, the attack signatures do not seek to detect a specific exploit, but rather protect against the vulnerability itself. This is important because attack exploits can vary, so writing signatures around a particular exploit is not a great tactic, but protecting against the actual vulnerability is much more powerful. Stage 8: IDP/IP actions Once an attack object in the IPS policy is matched, the SRX can execute an action on that specific session, along with actions on future sessions. The ability to execute an action on that particular session is known as an IDP action. IDP actions can be one of the following: No-Action, Drop-Packet, Drop-Connection, Close-Client, Close-Server, Close-Client-and-Server, DSCP-Marking, Recommended, or Ignore. IP actions are actions that can be enforced on future sessions. These actions include IP-Close, IP-Block, IP-Notify, and IP-Ratelimit.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Source: *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, pages 399-401. When assembled by the accused products, these components implement a variety of IDP processing algorithms. **Evidence '163 C1 1a(2)** **Application identification** Port-independent application identification enhances both security and manageability by eliminating the need to manually and comprehensively configure application-port mapping for the service objects and application objects used in the IDP rulebase and APE rulebase rules. Beginning with IDP OS Release 5.1, the application identification feature can match extended application signatures used in APE rulebase rules. Extended application signatures are also called nested application signatures. The predefined extended application signatures developed for IDP OS Release 5.1 include the most prevalent Web 2.0 applications running over HTTP. **User-defined application signatures** If the predefined signatures do not address all of your use cases, you can use the NSM Object Manager to create custom application signatures. **Application policy enforcement** The application policy enforcement (APE) rulebase enables you to mark, limit, or drop traffic that matches application signatures. **Application volume tracking** The application volume tracking (AVT) feature leverages Profiler functionality to collect statistics about application usage. **Multimethod attack detection** The IDP Series uses eight methods to detect malicious traffic. **Zero-day protection** The IDP rulebase attack objects detect protocol usages that violate published RFCs. Protocol anomaly detection protects your network from undiscovered vulnerabilities. **Protocol decoding** Juniper Networks Security Center (J-Security Center) provides a robust protocol detection engine that can decode more than 60 protocols and analyze and enforce proper usage in more than 500

	contexts. **Recommended security policy and predefined attack objects J**-Security Center provides a robust default security policy (called Recommended) and a comprehensive set of predefined attack objects (including those flagged as Recommended for various categories of attacks). The J-Security Center attack database includes more than 5500 signatures for identifying anomalies, attacks, spyware, and applications. **User-defined security policies and attack objects** If you choose, you can use the default security policy or other predefined templates as a basis for your own user-defined security policy. Similarly, you can use the predefined attack objects as a basis for your own user-defined attack objects. **Active response methods** J-Security Center attack objects are coded with recommended actions to take on the instant session, including drop packet, drop connection, close client, close server, and close client/server. You can rely on these or set your own. In addition, when the IDP Series device detects an attack from a particular IP address, it can block connections from the IP address for a configurable duration of time. **Passive response methods** The IDP Series supports several passive responses, including logging and TCP reset. **Traffic decryption and decapsulation** The IDP Series can decrypt or decapsulate traffic and then inspect the payload. We support decryption of SSL and decapsulation of GRE, GTP, IPsec ESP NULL, and MPLS traffic. **Stateful signature** The IDP rulebase attack object signatures are bound to protocol context. As a result, this detection method produces few false positives. **Protocol anomaly** The IDP rulebase attack objects detect protocol usages that violate published RFCs. This method protects your network from undiscovered vulnerabilities. **Traffic anomaly** The Traffic Anomalies rulebase uses heuristic rules to detect unexpected traffic patterns that might indicate reconnaissance or attacks. This method blocks distributed denial-of-service (DDoS) attacks and prevents reconnaissance activities.

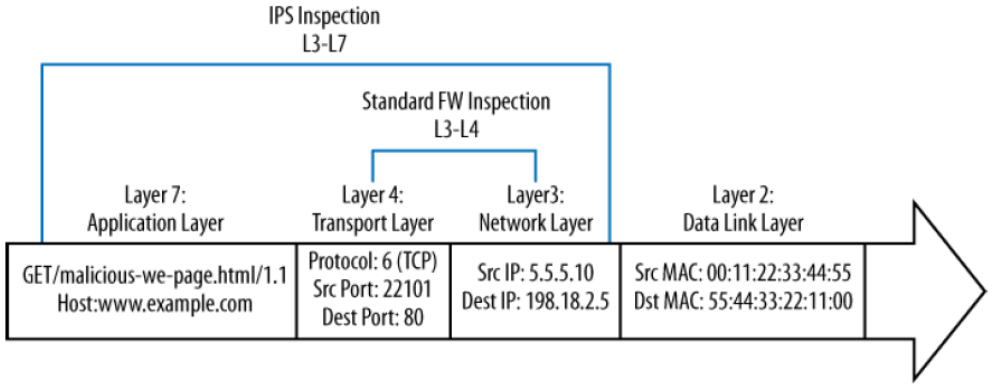
Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Backdoor The Backdoor rulebase uses heuristic-based anomalous traffic patterns and packet analysis to detect Trojans and rootkits. These methods prevent proliferation of malware in case other security measures have been compromised. **IP spoofing** The IDP Series device checks the validity of allowed addresses inside and outside the network, permitting only authentic traffic and blocking traffic with a disguised source. **Denial of service (DoS)** The SYN Protector rulebase provides two, alternative methods to prevent SYN-flood attacks. **Network honeypot** The IDP Series device impersonates vulnerable ports so you can track attacker reconnaissance activity. **Source:** *IDP Series, Concepts and Examples Guide,* Published by: Juniper Networks, Inc., February 2011, pages 3-7, http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf]
1b. for the first packet of the message, dynamically identifying a non-predefined sequence of components for processing the packets of the message such that the output format of the components of the nonpredefined sequence match the input format of the next component in the non-predefined sequence,	In the security functionality of the accused products, the first packet of a message flow is dynamically identified using deep packet inspection. Based on that inspection, the accused products utilize a technique of "policy expression", which are script-like directives that are loaded and re-loaded into the systems while they are running. They may be loaded and re-loaded into the systems by people, other systems or software, or both. The policies direct the system to identify the processing components and algorithms which are to be applied to the network traffic which is classified through the packet inspection. The accused products identify a packet (which implies a traffic/application flow), look at the latest loaded and resolved policy expression which applies to that traffic/application flow, and then arrange a sequence of processing components to affect the policy expression directive, the output format of which will match the input format of the next. Fully custom traffic/application flow specifications, as well as fully custom processing components, can be dynamically loaded and re-loaded into the system as well. Because of the configurability of policy expressions, traffic/applications specifications, there is an infinite set of resultant processing components – non-predefined – which will execute.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Evidence '163 C1 1b(1) Juniper's IPS protects the control plane and offers improved security for enhanced end-user experiences. We tightly integrate Junos OS IP technology with the most advanced security features, providing protection from a wide range of threats and attacks from both inside and outside the network, as well as supporting real-time policy assessment and enforcement. The Dynamic Application Awareness solution achieves these goals, providing the processing power for both stateful and stateless detection and identification of L4-L7 applications. Dynamic Application Awareness uses deep inspection (DI) technology to examine the L4-L7 payload via port, address, and signature detection methods. Residing on the MS-PIC in the M Series routers and on the MS-DPC in the MX Series routers; integrating IPS with M Series and MX Series routers. **Source:** *GENERATING NEW REVENUE STREAMS AND INCREASING NETWORK SECURITY Dynamic Application Awareness and Intrusion Prevention System*, Published by Juniper Networks, Dec, 2009, www.**juniper**.net/us/en/local/pdf/whitepapers/2000339-en.pdf **Evidence '163 C1 1b(2)** At a high level, IPS works by scrutinizing all of the bits contained within packets to look for both known and unknown attacks. Traditional firewalls primarily look only at Layers 3 and 4 when it comes to security, and ignore the actual contents of the payloads themselves.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case



Firewall inspection of attack versus IPS

Source: *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, page 391

Evidence '163 C1 1b(3)

Configuration settings that determine the fate of a packet—such as the security policy that applies to it, if it requires an Application Layer Gateway (ALG), if NAT is applied to translate the packet's source and/or destination IP address—are assessed for the first packet of a flow. To determine if a flow exists for a packet, the NPU attempts to match the packet's information to that of an existing session based on the following match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Unique session token number for a given zone and virtual router

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	The security policy to be used for the first packet of a flow is cached in a flow table for use with the same flow and closely related flows. Security policies are associated with zones. A zone is a collection of interfaces that define a security boundary. A packet's incoming zone, as determined by the interface through which it arrived, and its outgoing zone, as determined by the forwarding lookup, together determines which policy is used for packets of the flow. **Source:** *Junos OS Security Configuration Guide,* Published by Juniper Networks, Inc., March 2011, pages 5-6, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf **DEEP INSPECTION** **Evidence '163 C1 1b(4)** A screen is a built-in tunable protection mechanism that performs a variety of security functions to keep the network safe. Screens are extremely efficient and can be tuned to operate in a small enterprise or in the largest carrier networks. Screens are widely used to add additional protections both at the edge of the network and to internal segments to protect the network from attacks and internal misconfigurations that could impact network availability. Screens are good at detecting and preventing many types of malicious traffic. Screen checks take place very early in packet processing to make mitigation as efficient and fast as possible. Although they take more processing power than a firewall filter, they are able to look deeper into the packet and at the entire session flow, essentially enabling the SRX to block very large and complex attacks. On the higher-end SRX models, many of these screens are handled in hardware, so the traffic is dropped extremely close to the ingress interface. You may notice that the screen checks take place on both the slow path and the fast path. Once a session is permitted by policy and is established, the SRX continues to monitor that connection for signs of any malicious traffic or flooding beyond its preconfigured thresholds. If it sees any malicious traffic, it blocks and drops the packets.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

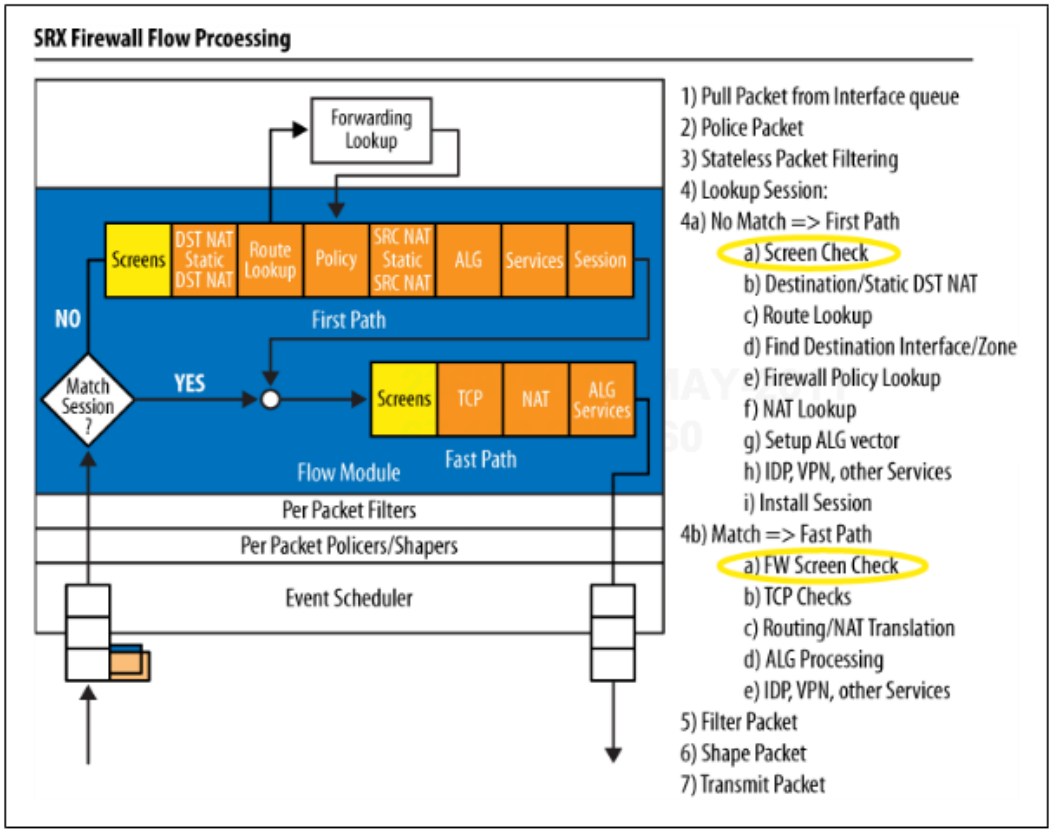


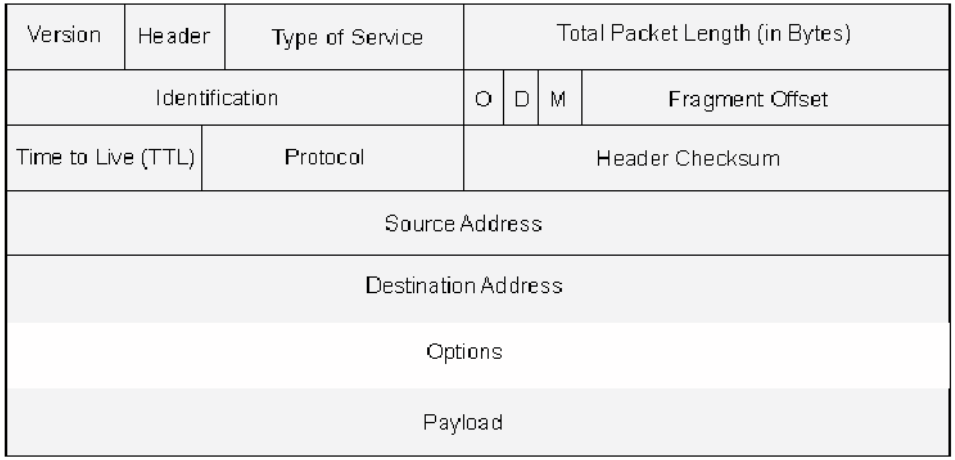
Figure 7-2. Where screen checks take place in the SRX packet flow

[from *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, page 346]

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

Evidence '163 C1 1b(5)

RFC 791 states that these options are "unnecessary for the most common communications" and, in reality, they rarely appear in IP packet headers. These options appear after the destination address in an IP packet header, as shown (here). When they do appear, they are frequently being put to some illegitimate use:



If a packet with any of the previous IP options is received, Junos OS flags this as a network reconnaissance attack and records the event for the ingress interface.

This example shows how to detect packets that use IP screen options for reconnaissance.

Source: *Junos OS Security Configuration Guide*, Published by Juniper Networks, May 2010, Page 1025, 1028, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

The accused products include a generalized mechanism for doing packet inspection/flow classification.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Evidence '163 C1 1b(6) Application identification supports user-defined custom application signatures for applications and nested applications. With custom application signatures, you can create signatures that will detect applications that are not part of the predefined application package. **Source:** *Junos OS Security Configuration Guide, Juniper Networks, May 2010, Page 1025, 1028,* https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf The accused products also support inspection/classification of encapsulated or encrypted traffic. **Evidence '163 C1 1b(7)** Generic Routing Encapsulation (GRE) is a tunneling protocol designed to encapsulate a wide variety of network layer packets inside IP tunneling packets. The original packet is the payload for the final packet. The protocol is used on the Internet to secure virtual private networks. To inspect the payload of an encapsulated packet, the IDP process engine must decapsulate it. IDP Series devices support decapsulation for IP-in-GRE and PPP-in-GRE. GPRS Tunneling Protocol (or GTP) is an IP-based protocol used within Global System for Mobile communication (GSM) and Universal Mobile Telecommunications System (UMTS) networks. To inspect the payload of an encapsulated traffic, the IDP process engine must decapsulate it. IDP Series devices support decapsulation for UDP GTPv0 and GTPv1. Internet Protocol Security (IPsec) virtual private networks use the Encapsulated Security Payload (ESP) protocol and the NULL encryption algorithm to ensure the authenticity, integrity, and confidentiality of IP packets. To inspect the payload of an encapsulated packet, the IDP process engine must decapsulate it. IDP Series devices support decapsulation for IPsec ESP NULL traffic. Multiprotocol Label Switching (MPLS) is an IP label switching technology that enables predetermined paths to specific destinations, called Label Switched Paths (LSPs), to be established through an inherently

	connectionless IP network. In MPLS networks, packets contain short labels that describe how to forward them through the network. With MPLS decapsulation enabled, the IDP engine can inspect the IPv4 payload and pass through non-IPv4 payload. Secure Sockets Layer (SSL) is a cryptographic protocol that adds security to TCP/IP communication. Several versions of the SSL and Transport Layer Security (TLS) protocols are in widespread use in applications like Web browsing, electronic mail, Internet faxing, instant messaging, and voice over IP (VoIP). SSL and TLS encrypt the Transport Layer protocol datagrams that carry the payload of these communications. While encryption is an excellent way to keep private data from prying eyes, without inspection by the IDP Series device, it also unwittingly opens a network to dangerous viruses, trojans, or network attacks. To inspect the HTTP payload of HTTPS traffic, the IDP Series device must decrypt the HTTPS session. Your security policy can examine both the SSL session and the decrypted HTTP payload. **Source:** *IDP Series Concepts and Examples Guide, Juniper Networks, Published Feb. 2011*, http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, pages 177-179 **Evidence '163 C1 1b(8)** To secure your network from new viruses and attacks, your security solution must offer multiple attack detection methods and an efficient way to use the various capabilities. To stay one step ahead of these attacks, you need a solution that can adapt to ever-changing security threats and allow you to do so with minimal effort. Juniper Networks IDP Series Intrusion Detection and Prevention Appliances with Multi-Method Detection (MMD), offers comprehensive coverage by leveraging multiple detection mechanisms. For example, by utilizing signatures, as well as other detection methods including protocol anomaly traffic anomaly detection, the Juniper Networks IDP Series appliances can thwart known attacks as well as possible future variations of the attack. Backed by Juniper Networks Security Lab, signatures for detection of new attacks are generated on a daily

	basis, working very closely with many software vendors. While an IDP solution is a critical component of every enterprise security infrastructure, it also offers the benefit of streamlining your business based on the applications used in the network. In addition to identifying viruses and attacks, the Juniper Networks IDP Series can identify the application associated with the particular traffic. Application intelligence enables accurate detection and reporting of volume used by applications such as social networking, peer-to-peer, or instant messaging. Armed with the knowledge of these applications running in the network, administrators can easily manage them by limiting bandwidth, restricting their use, or changing their prioritization for the best network optimization. By accurately identifying and prioritizing application traffic, enterprises can ensure the necessary network bandwidth for business-critical applications without banning or blocking non-business applications. If necessary, specific application traffic can be blocked altogether to meet business or regulatory compliance. **Source:** *IDP Series Intrusion Detection and Prevention Appliances*, published by Juniper Networks, Oct 2009, http://www.juniper.net/us/en/local/pdf/brochures/1500025-en.pdf] **Evidence '163 C1 1b(9)** A security policy, which can be configured from the user interface, controls the traffic flow from one zone to another zone by defining the kind(s) of traffic permitted from specified IP sources to specified IP destinations at scheduled times. Policies allow you to **deny, permit, reject** (deny and send a TCP RST or ICMP port unreachable message to the source host), **encrypt** and **decrypt**, **authenticate, prioritize, schedule, filter**, and **monitor** the traffic attempting to cross from one security zone to another. You decide which users and what data can enter and exit, and **when and where** they can go. **Source:** *Junos OS Security Configuration Guide, Juniper Networks, May 2010, Page 146,* https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Evidence '163 C1 1b(10) There can be many dozens —or even thousands— of policies configured in various SRX devices (this number varies by platform). When packets ingress any of the devices, they are evaluated against security policies. If a match is found then the SRX does what it was instructed to do with those packets and stops evaluating through the rest of the policies. Security policies are at the heart of any of the firewall functions of the SRX Services Gateway platform. By default, traffic entering an interface destined to any address is going to be blocked. This is the expected default behavior, and no traffic is allowed through the SRX until you permit it to enter by using security policies. Policy configuration entitles an IF-THEN-ELSE algorithm: IF traffic X is matched, THEN action Y is performed, ELSE drop packet (default behavior). Matching traffic (IF statement) consists of looking at packets for the five following elements: 1. Source zone: the predefined or custom zone created from the perspective of the SRX that you are configuring. 2. Source IP: any IP address, or an address book, that specifies a host IP, or a subnet. The source selected has to match the source zone. 3. Destination zone: predefined or custom zone created from the perspective of the SRX that you are configuring. 4. Destination IP: any IP address, or an address book that specifies a host IP, or a subnet. The destination selected has to match the destination zone. 5. Application: predefined or custom service that defines the source/destination ports, protocol involved, and timeout value. If an incoming packet matches all the previous five elements, the action (THEN statement) defines what to do **with this or any other packets matching the same combination**:

- deny: drops the packet (silently).
- reject: drops the packet and sends a TCP-Reset to the originator of the traffic.
- permit: permits the packet.
- log: instructs the SRX to create a log entry for matching packets.
- count: provides accounting information per session.

Source: *Day One: Deploying SRX Series Services Gateways*, *Junos Dynamic Services Series*, published by Juniper Networks, Jan 2011, pages 54, 55, http://www.juniper.net/us/en/community/junos/training-certification/day-one/dynamic-services-series/deploying-srx-series/

The accused products not only support the "firewall" types of policies mentioned above, but they support much more complicated IDP policies. IDP policies are sometimes called "rulebases" and the traffic classification specification used to match a rulebase is often called a "signature" to reflect their more general programmability.

Evidence '163 C1 1b(11)

To help block malicious application-level attacks, Juniper Networks seamlessly integrates intrusion prevention across the entire product line. For central enterprise sites, data center environments and service provider networks with high volumes of throughput, the Juniper Networks ISG Series Integrated Security Gateways with IPS, Juniper Networks SRX100, SRX210, SRX220, SRX240, SRX650, SRX1400, SRX3000 line and SRX5000 line of services gateways can be deployed for application-level protection. The ISG Series and SRX Series tightly integrates the same software found on the Juniper Networks IDP Series Intrusion Detection and Prevention Appliances to provide unmatched application-level protection against worms, trojans, spyware, and malware. More than 60 protocols are recognized including those used by advanced applications such as VoIP and streaming media.

Unmatched security processing power and network segmentation features protect critical high-speed networks against the penetration and proliferation of existing and emerging application-level threats. With multiple attack detection mechanisms, including stateful signatures and protocol anomaly, the ISG Series and SRX Series Services Gateways performs in-depth analysis of application protocol, context, state and behavior to deliver Zero-day protection.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Security administrators can deploy Juniper Networks AppSecure capability using deep inspection to block application-level attacks before they infect the network and inflict any damages. AppSecure utilizes advanced, high-performance detection mechanisms integrated with stateful inspection firewall, along with multiple threat inspection engines operating in parallel to accurately detect advanced persistent threats, including those found in nested applications within applications. **Source:** *Integrated Firewall/VPN Platforms*, published by Juniper Networks, Nov. 2010, http://www.juniper.net/us/en/local/pdf/brochures/1500024-en.pdf **Evidence '163 C1 1b(12)** The IDP rulebase employs an attack object database to support two robust detection methods: stateful signatures and protocol anomalies. A stateful signature combines an attack pattern with service, context, and other properties into a signature attack object. As a result, the IDP system does not need to expend resources inspecting huge sections of network traffic where attacks cannot possibly be, and IDP produces very few false positives. A protocol anomaly is a deviation from protocol standards established by the Internet Engineering Taskforce (IETF) Request for Comment (RFC) process. Traffic that does not adhere to these standards is suspicious because most legitimate applications adhere to the standards, and anomalies can fairly be regarded as purposeful attempts to evade an intrusion detection system (IDS). IDP protocol-anomaly attack objects find traffic that deviates from IETF RFC standards. When you create rules for the IDP rulebase, you specify: • A source/destination/service match condition • Attack objects • Action • Notification options **The IDP engine inspects the session beginning with the first packet to determine whether the session**

matches a rule. If the session matches all rule settings for source, destination, service, and VLAN tag ID, the IDP system decodes the traffic and inspects the session packets for the attack objects specified in the rule.

Source: *IDP Series Concepts and Examples Guide, Juniper Networks, Published Feb. 2011,* http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, pages 91, 92

Unified Threat Management (UTM) is a term used to describe the consolidation of several security features into one device, protecting against multiple threat types.

Evidence '163 C1 1b(13)

The security features provided as part of the UTM solution are:

- **Antispam**—E-mail spam consists of unwanted e-mail messages, usually sent by commercial, malicious, or fraudulent entities. The antispam feature examines transmitted e-mail messages to identify e-mail spam. When the device detects an e-mail message deemed to be spam, it either drops the message or tags the message header or subject field with a preprogrammed string. The antispam feature uses a constantly updated spam block list (SBL). Sophos [n.b., an outside company accessed through an algorithm which goes to a special internet site], updates and maintains the IP-based SBL.
- **Full File-Based Antivirus**—A virus is executable code that infects or attaches itself to other executable code to reproduce itself. Some malicious viruses erase files or lock up systems. Other viruses merely infect files and overwhelm the target host or network with bogus data. The full file-based antivirus feature provides file-based scanning on specific Application Layer traffic checking for viruses against a virus signature database. It collects the received data packets until it has reconstructed the original application content, such as an e-mail file attachment, and then scans this content.
- **Express Antivirus**—Express antivirus scanning is offered as a less CPU intensive alternative to the full file-based antivirus feature. The express antivirus feature, like the full antivirus feature, scans specific Application Layer traffic for viruses against a virus signature database. However, unlike full antivirus, express antivirus does not reconstruct the original application content. Rather, it just sends (streams) the received data packets, as is, to the scan engine. With express antivirus, the virus scanning is executed by a hardware pattern matching engine. This improves performance while scanning is occurring, but the level

of security provided is lessened.

- **Content Filtering**—Content filtering blocks or permits certain types of traffic based on the MIME type, file extension, protocol command, and embedded object type.
- **Web Filtering**—Web filtering lets you manage Internet usage by preventing access to inappropriate Web content.

Before you can configure most UTM features, you must first configure the custom objects for the feature in question. Custom objects are global parameters for UTM features. This means that configured custom objects can be applied to all UTM policies where applicable, rather than only to individual policies.

Source: *Junos OS Security Configuration Guide, Juniper Networks, May 2010, Page 843-844,* https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

Evidence '163 C1 1b(14)

Configuration

The unified threat management [UTM) implementation in Junos OS leverages security policies as a central point where traffic is classified and directed to the appropriate modules for processing. In practice, a UTM policy specifying all UTM-related parameters is attached to a security policy, and matching traffic is processed by the UTM module according to the configuration of the UTM policy.

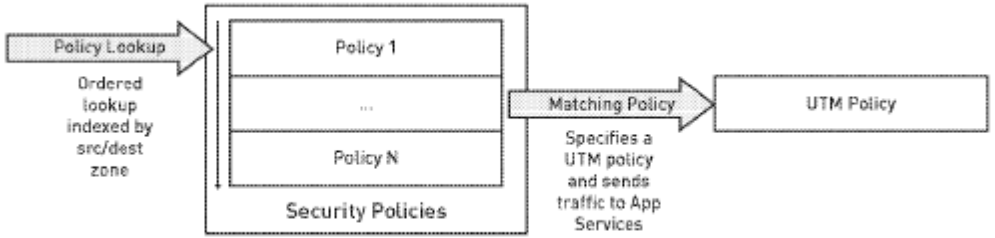


Figure 1: UTM policies

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

In a similar fashion, a UTM policy ties a set of protocols to one or multiple feature profiles. Each feature profile determines the specific configuration for each feature (antivirus, content filtering, anti-spam).

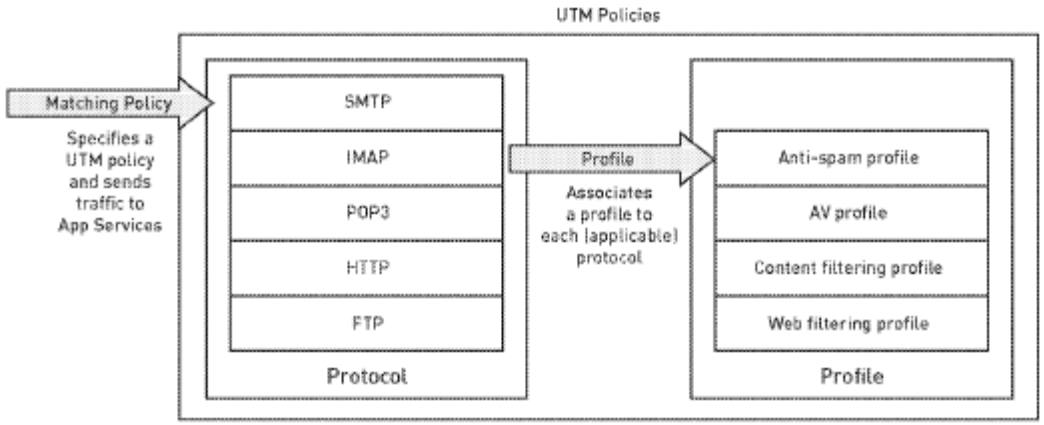


Figure 2: UTM policies and feature profiles

Source: "*Application Note:* Content Filtering For Branch SRX Series and J Series", JUNIPER01475161

Evidence '163 C1 1b(15)

Once a security policy specifies a UTM policy, a transparent proxy processes all matching traffic and, in the case of this book's SRX devices, modifies the contents of the traffic or generate error messages back to the user. To proxy a session, an SRX device acts both as a TCP client and as a server terminating and originating a TCP session. This uses significant resources, in terms of both memory and CPU, which puts some constraints on the total number of sessions an SRX can proxy (and, in turn, the total number of concurrent sessions using UTM features). The TCP proxy code feeds a data stream to the protocol parser which, in turn, can decode the protocols supported by UTM, namely FTP, HTTP, SMTP, POP3, and IMAP. The protocol parser extracts the relevant content from each protocol and sends it to the appropriate engine for processing, all of which is depicted in Figure 9-1.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

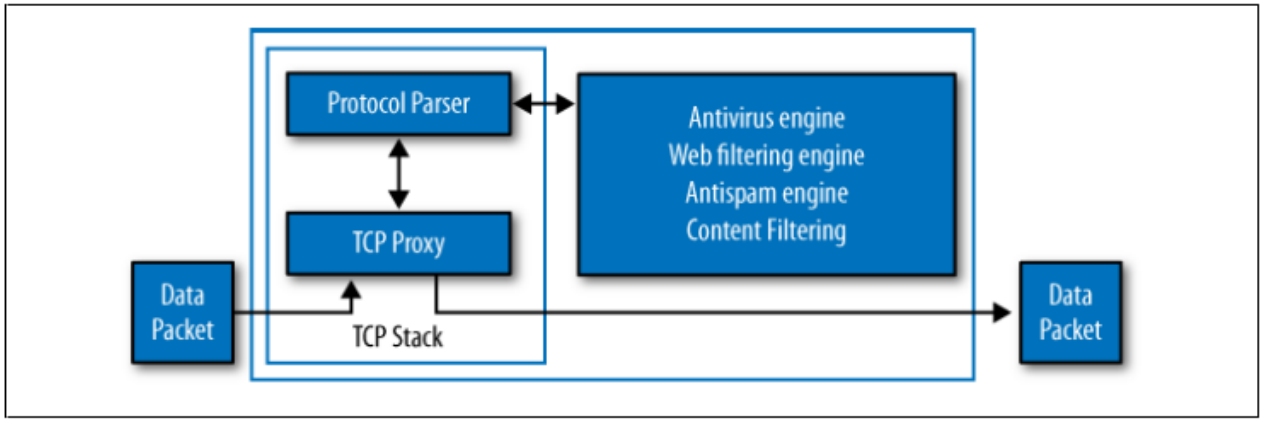


Figure 9-1. How SRX proxies a session

Source: *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, page 489

The accused products are based on an architecture which Juniper calls "the Dynamic Services Architecture". This architecture dynamically arranges and connects the needed components to implement the security processing identified first by the classification, and then by the policy directives.

Evidence '163 C1 1b(16)

As opposed to most appliances that must examine every packet in every session, Dynamic Application Awareness and IPS enable you to identify applications by optionally configuring the software to examine just the first few packets of newly initiated sessions. Once the application is identified, a router-integrated policy manager provisions the forwarding plane (in real time) with the appropriate session handling instructions (such as, block, rate limit, apply CoS, etc).

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	The forwarding plane resources then ensure that the session is treated and forwarded in accordance with the policy, and the service plane resources can be allocated to other sessions, permitting the solution to scale with high performance. Traffic flows through the Dynamic Application Awareness and the IPS processes as follows (Figure 3). 1. The subscriber initiates a session. 2. Dynamic Application Awareness: The session is forwarded to the Dynamic Application Awareness engine hosted on the MS-PIC/MS-DPC. IPS: The session is forwarded to the IPS engine hosted on the MS-PIC/MS-DPC. 3. Dynamic Application Awareness: The packet header is searched to identify the application based on its port, address, or signature. IPS: The packet is searched to identify threats and attacks using the following detection mechanisms. • Anomaly—check traffic against protocol standard. • Signature—protocol-aware context signature. • Backdoor—detect traffic bypassing normal authentication procedures. 4. The application policy request is forwarded to a local policy manager. 5. The local policy manager compares the identified application against a customer-defined list of application handling instructions.By default, all packets in the session are examined. One user-configurable option is that the session incurs no further analysis. In this case, Dynamic Application Awareness or IPS no longer analyzes this session, and its resources are available to analyze other sessions. Otherwise, the traffic is pushed to the forwarding plane (step 6). **6. The local policy manager provisions the appropriate enforcement functions on the forwarding plane in real time.** • Rate limit traffic, packet drop • Classify traffic (DSCP mark for CoS handling) • Connection close, block traffic

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

• Statistic gathering and logging

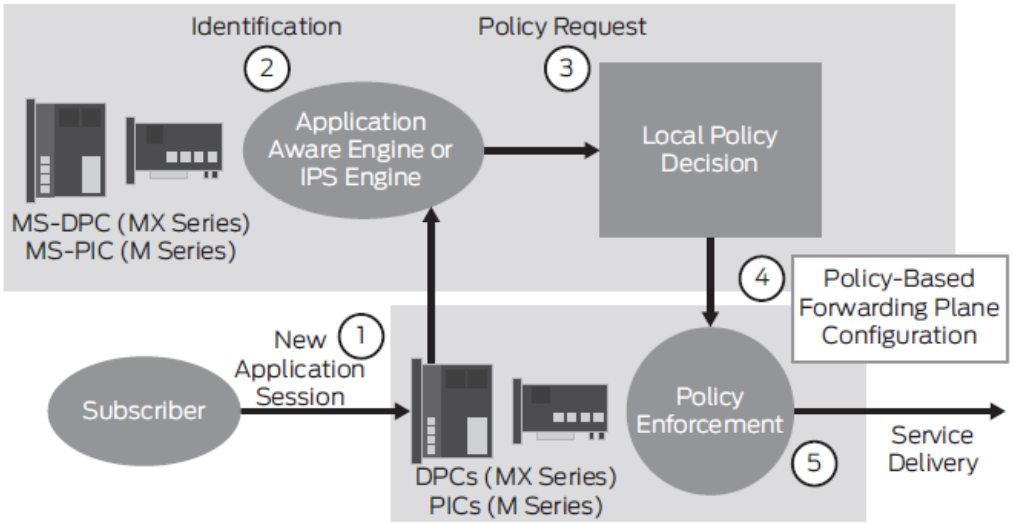


Figure 3: Logical packet flow

Source: *GENERATING NEW REVENUE STREAMS AND INCREASING NETWORK SECURITY Dynamic Application Awareness and Intrusion Prevention System*, Published by Juniper Networks, Dec, 2009, www.**juniper**.net/us/en/local/pdf/whitepapers/2000339-en.pdf

Evidence '163 C1 1b(17)

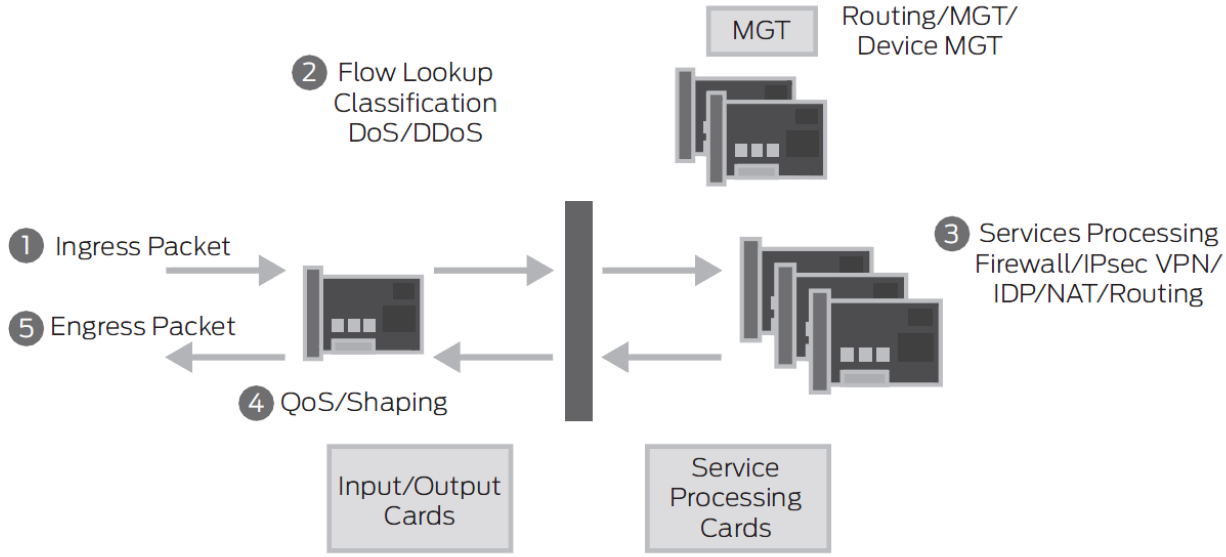
Not Just Another Chassis Design

The Dynamic Services Architecture is based on a chassis design; however, it is a complete departure from traditional chassis architecture. Rather than simply providing a fast backplane, the Dynamic Services

	Architecture includes the management and control necessary to incorporate individual blades into a powerful collective solution. Rather than housing disparate cards, the Dynamic Services Architecture adds each blade into a growing pool of resources. These resources can then be utilized as necessary for optimal processing of network traffic. **Switch Fabric, Control Board and Route Engine** At the heart of the Dynamic Services Architecture is the Switch fabric and Control Board (SCB). The SCB transforms the chassis from a simple blade enclosure into a highly effective mesh network. The purpose of the SCB is to allow all blades in the chassis to send traffic at extremely high bandwidth. The Route Engine (RE) is tightly coupled with the functionality of the SCB and can be considered the central nervous system of the architecture. The RE is the control plane of the chassis and provides overall management and communications to and from system administrators, as well as calculating route tables for routing network traffic. The operating system, which includes key chassis functionality, also runs on the RE. In the case of networking and security, functionalities such as advanced routing, switching, flow-base security, zone-based management, and screens are available on the OS. **Service Processing Cards** If the RE is the central nervous system of the chassis, the Service Processing Card (SPC), is the brain. SPCs are blades that provide the capacity to perform the heavy lifting of processing network packets. **Session Distribution** The Dynamic Services Architecture also supports automatic load balancing with advanced performance and capacity due to its session distribution design. This is enabled by the intelligent input/output and network processing subsystems, which balance sessions across the shared pool of SPCs (the "brain" discussed above). This is possible because all the SPCs in the system run the same services and have the same configuration. There is no specific mapping from one IOC to one SPC; rather, each flow is mapped dynamically upon

	session creation. Any session coming in any port can be forwarded, on a session-by-session basis, to any SPC in the system. This load balancing is performed automatically, with no configuration or oversight by the system administrator. This is dramatically opposed to traditional chassis-based solutions, where each processing blade is an independent firewall, with its own dedicated traffic, unique configuration, and routing support. **Packet Flow** In the same way, the packet flow within the Dynamic Services Architecture becomes fully integrated and far easier to manage. No longer is it necessary for administrators to provide separate instructions to each blade for traffic management. Each packet traversing the system now takes the same basic path: 1. The ingress packet enters Ethernet port on the IOC. 2. It is processed by the IOC and passed to the switch fabric. 3. One processing unit on the SPC receives and processes the packet for the firewall, IPsec VPN, and/or IPS. If the packet is to be dropped, the SPC does so and will typically log the event. 4. If the packet is to be passed, it is passed back through the switch fabric to the IOC, where it is processed by the IOC processor, where QoS is applied if necessary. 5. The packet is then passed out the Ethernet port to egress the system.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	 Figure 2: An example of a fully integrated packet flow (SRX5000 line) **Source:** *Dynamic Services Architecture: a Revolutionary Approach to Integrated Network Security*, published by Juniper Networks, Oct 2009 , pages 4-6, https://www.juniper.net/us/en/local/pdf/whitepapers/2000288-en
1c. wherein dynamically identifying includes selecting individual components to create the nonpredefined sequence of components after the first packet is received;	The accused products utilize a technique of "policy expression", which are script-like directives that are loaded and re-loaded into the systems while they are running. They may be loaded and re-loaded into the systems by people, other systems or software, or both. The policies direct the system to identify the processing components and algorithms which are to be applied to the network traffic which is classified through the packet inspection. The accused products identify a packet (which implies a traffic/application flow), look at the latest loaded and resolved policy expression which applies to that traffic/application flow, and then arrange a sequence of processing components to affect the policy expression directive. The system will contain a large library of processing

components. Fully custom traffic/application flow specifications, as well as fully custom processing components, can be dynamically loaded and re-loaded into the system as well. Because of the configurability of policy expressions, traffic/applications specifications, there is an infinite set of resultant processing components – non-predefined – which will execute.

Evidence '163 C1 1c(1)

To secure your network from new viruses and attacks, your security solution must offer multiple attack detection methods and an efficient way to use the various capabilities.

To stay one step ahead of these attacks, you need a solution that can adapt to ever-changing security threats and allow you to do so with minimal effort.

Juniper Networks IDP Series Intrusion Detection and Prevention Appliances with Multi-Method Detection (MMD), offers comprehensive coverage by leveraging multiple detection mechanisms. For example, by utilizing signatures, as well as other detection methods including protocol anomaly traffic anomaly detection, the Juniper Networks IDP Series appliances can thwart known attacks as well as possible future variations of the attack.

Backed by Juniper Networks Security Lab, signatures for detection of new attacks are generated on a daily basis, working very closely with many software vendors.

While an IDP solution is a critical component of every enterprise security infrastructure, it also offers the benefit of streamlining your business based on the applications used in the network. In addition to identifying viruses and attacks, the Juniper Networks IDP Series can identify the application associated with the particular traffic. Application intelligence enables accurate detection and reporting of volume used by applications such as social networking, peer-to-peer, or instant messaging. Armed with the knowledge of these applications running in the network, administrators can easily manage them by limiting bandwidth, restricting their use, or changing their prioritization for the best network optimization.

By accurately identifying and prioritizing application traffic, enterprises can ensure the necessary network bandwidth for business-critical applications without banning or blocking non-business applications. If

	necessary, specific application traffic can be blocked altogether to meet business or regulatory compliance. **Source:** *IDP Series Intrusion Detection and Prevention Appliances*, published by Juniper Networks, Oct 2009, http://www.juniper.net/us/en/local/pdf/brochures/1500025-en.pdf] **Evidence '163 C1 1c(2)** A security policy, which can be configured from the user interface, controls the traffic flow from one zone to another zone by defining the kind(s) of traffic permitted from specified IP sources to specified IP destinations at scheduled times. Policies allow you to **deny, permit, reject** (deny and send a TCP RST or ICMP port unreachable message to the source host), **encrypt** and **decrypt**, **authenticate, prioritize, schedule, filter**, and **monitor** the traffic attempting to cross from one security zone to another. You decide which users and what data can enter and exit, and **when and where** they can go. **Source:** *Junos OS Security Configuration Guide, Juniper Networks, May 2010, Page 146,* https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf **Evidence '163 C1 1c(3)** There can be many dozens —or even thousands— of policies configured in various SRX devices (this number varies by platform). When packets ingress any of the devices, they are evaluated against security policies. If a match is found then the SRX does what it was instructed to do with those packets and stops evaluating through the rest of the policies. Security policies are at the heart of any of the firewall functions of the SRX Services Gateway platform. By default, traffic entering an interface destined to any address is going to be blocked. This is the expected default behavior, and no traffic is allowed through the SRX until you permit it to enter by using security

	policies. Policy configuration entitles an IF-THEN-ELSE algorithm: IF traffic X is matched, THEN action Y is performed, ELSE drop packet (default behavior). Matching traffic (IF statement) consists of looking at packets for the five following elements: 6. Source zone: the predefined or custom zone created from the perspective of the SRX that you are configuring. 7. Source IP: any IP address, or an address book, that specifies a host IP, or a subnet. The source selected has to match the source zone. 8. Destination zone: predefined or custom zone created from the perspective of the SRX that you are configuring. 9. Destination IP: any IP address, or an address book that specifies a host IP, or a subnet. The destination selected has to match the destination zone. 10. Application: predefined or custom service that defines the source/destination ports, protocol involved, and timeout value. If an incoming packet matches all the previous five elements, the action (THEN statement) defines what to do **with this or any other packets matching the same combination**: • deny: drops the packet (silently). • reject: drops the packet and sends a TCP-Reset to the originator of the traffic. • permit: permits the packet. • log: instructs the SRX to create a log entry for matching packets. • count: provides accounting information per session. **Source:** *Day One: Deploying SRX Series Services Gateways, Junos Dynamic Services Series*, published by Juniper Networks, Jan 2011, pages 54, 55, http://www.juniper.net/us/en/community/junos/training-certification/day-one/dynamic-services-series/deploying-srx-series/ The accused products not only support the "firewall" types of policies mentioned above, but they support much more

complicated IDP policies. IDP policies are sometimes called "rulebases" and the traffic classification specification used to match a rulebase is often called a "signature" to reflect their more general programmability.

Evidence '163 C1 1c(4)

To help block malicious application-level attacks, Juniper Networks seamlessly integrates intrusion prevention across the entire product line. For central enterprise sites, data center environments and service provider networks with high volumes of throughput, the Juniper Networks ISG Series Integrated Security Gateways with IPS, Juniper Networks SRX100, SRX210, SRX220, SRX240, SRX650, SRX1400, SRX3000 line and SRX5000 line of services gateways can be deployed for application-level protection. The ISG Series and SRX Series tightly integrates the same software found on the Juniper Networks IDP Series Intrusion Detection and Prevention Appliances to provide unmatched application-level protection against worms, trojans, spyware, and malware. More than 60 protocols are recognized including those used by advanced applications such as VoIP and streaming media.

Unmatched security processing power and network segmentation features protect critical high-speed networks against the penetration and proliferation of existing and emerging application-level threats. With multiple attack detection mechanisms, including stateful signatures and protocol anomaly, the ISG Series and SRX Series Services Gateways performs in-depth analysis of application protocol, context, state and behavior to deliver Zero-day protection.

Security administrators can deploy Juniper Networks AppSecure capability using deep inspection to block application-level attacks before they infect the network and inflict any damages. AppSecure utilizes advanced, high-performance detection mechanisms integrated with stateful inspection firewall, along with multiple threat inspection engines operating in parallel to accurately detect advanced persistent threats, including those found in nested applications within applications.

Source: *Integrated Firewall/VPN Platforms*, published by Juniper Networks, Nov. 2010, http://www.juniper.net/us/en/local/pdf/brochures/1500024-en.pdf

Evidence '163 C1 1c(5)

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	The IDP rulebase employs an attack object database to support two robust detection methods: stateful signatures and protocol anomalies. A stateful signature combines an attack pattern with service, context, and other properties into a signature attack object. As a result, the IDP system does not need to expend resources inspecting huge sections of network traffic where attacks cannot possibly be, and IDP produces very few false positives. A protocol anomaly is a deviation from protocol standards established by the Internet Engineering Taskforce (IETF) Request for Comment (RFC) process. Traffic that does not adhere to these standards is suspicious because most legitimate applications adhere to the standards, and anomalies can fairly be regarded as purposeful attempts to evade an intrusion detection system (IDS). IDP protocol-anomaly attack objects find traffic that deviates from IETF RFC standards. When you create rules for the IDP rulebase, you specify: • A source/destination/service match condition • Attack objects • Action • Notification options **The IDP engine inspects the session beginning with the first packet to determine whether the session matches a rule**. If the session matches all rule settings for source, destination, service, and VLAN tag ID, the IDP system decodes the traffic and inspects the session packets for the attack objects specified in the rule. **Source:** *IDP Series Concepts and Examples Guide, Juniper Networks, Published Feb. 2011,* http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, pages 91, 92 Unified Threat Management (UTM) is a term used to describe the consolidation of several security features into one device, protecting against multiple threat types. **Evidence ‘163 C1 1c(6)**

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	The security features provided as part of the UTM solution are: • **Antispam**—E-mail spam consists of unwanted e-mail messages, usually sent by commercial, malicious, or fraudulent entities. The antispam feature examines transmitted e-mail messages to identify e-mail spam. When the device detects an e-mail message deemed to be spam, it either drops the message or tags the message header or subject field with a preprogrammed string. The antispam feature uses a constantly updated spam block list (SBL). Sophos [n.b., an outside company accessed through an algorithm which goes to a special internet site], updates and maintains the IP-based SBL. • **Full File-Based Antivirus**—A virus is executable code that infects or attaches itself to other executable code to reproduce itself. Some malicious viruses erase files or lock up systems. Other viruses merely infect files and overwhelm the target host or network with bogus data. The full file-based antivirus feature provides file-based scanning on specific Application Layer traffic checking for viruses against a virus signature database. It collects the received data packets until it has reconstructed the original application content, such as an e-mail file attachment, and then scans this content. • **Express Antivirus**—Express antivirus scanning is offered as a less CPU intensive alternative to the full file-based antivirus feature. The express antivirus feature, like the full antivirus feature, scans specific Application Layer traffic for viruses against a virus signature database. However, unlike full antivirus, express antivirus does not reconstruct the original application content. Rather, it just sends (streams) the received data packets, as is, to the scan engine. With express antivirus, the virus scanning is executed by a hardware pattern matching engine. This improves performance while scanning is occurring, but the level of security provided is lessened. • **Content Filtering**—Content filtering blocks or permits certain types of traffic based on the MIME type, file extension, protocol command, and embedded object type. • **Web Filtering**—Web filtering lets you manage Internet usage by preventing access to inappropriate Web content. Before you can configure most UTM features, you must first configure the custom objects for the feature in question. Custom objects are global parameters for UTM features. This means that configured custom objects can be applied to all UTM policies where applicable, rather than only to individual policies. **Source:** *Junos OS Security Configuration Guide, Juniper Networks, May 2010, Page 843-844,* https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-

security/junos-security-swconfig-security.pdf

Evidence '163 C1 1c(7)

Once a security policy specifies a UTM policy, a transparent proxy processes all matching traffic and, in the case of this book's SRX devices, modifies the contents of the traffic or generate error messages back to the user. To proxy a session, an SRX device acts both as a TCP client and as a server terminating and originating a TCP session. This uses significant resources, in terms of both memory and CPU, which puts some constraints on the total number of sessions an SRX can proxy (and, in turn, the total number of concurrent sessions using UTM features). The TCP proxy code feeds a data stream to the protocol parser which, in turn, can decode the protocols supported by UTM, namely FTP, HTTP, SMTP, POP3, and IMAP. The protocol parser extracts the relevant content from each protocol and sends it to the appropriate engine for processing, all of which is depicted in Figure 9-1.

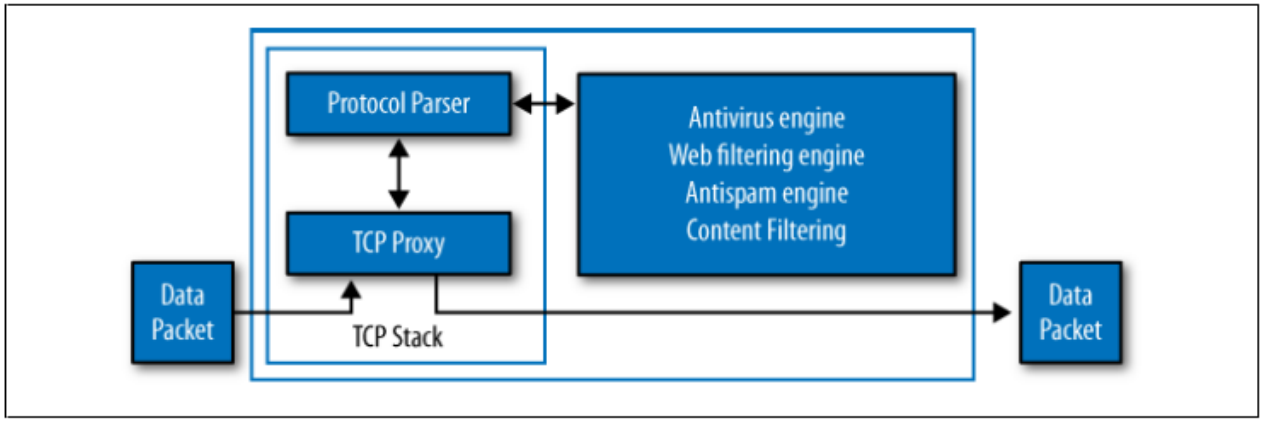


Figure 9-1. How SRX proxies a session

Source: *Junos Security*, By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010,

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

ISBN-13: 978-1-4493-8171-4, page 489

The accused products are based on an architecture which Juniper calls "the Dynamic Services Architecture". This architecture dynamically arranges and connects the needed components to implement the security processing identified first by the classification, and then by the policy directives.

Evidence '163 C1 1c(8)

As opposed to most appliances that must examine every packet in every session, Dynamic Application Awareness and IPS enable you to identify applications by optionally configuring the software to examine just the first few packets of newly initiated sessions. Once the application is identified, a router-integrated policy manager provisions the forwarding plane (in real time) with the appropriate session handling instructions (such as, block, rate limit, apply CoS, etc).

The forwarding plane resources then ensure that the session is treated and forwarded in accordance with the policy, and the service plane resources can be allocated to other sessions, permitting the solution to scale with high performance. Traffic flows through the Dynamic Application Awareness and the IPS processes as follows (Figure 3).

7. The subscriber initiates a session.

8. Dynamic Application Awareness: The session is forwarded to the Dynamic Application Awareness engine hosted on the MS-PIC/MS-DPC. IPS: The session is forwarded to the IPS engine hosted on the MS-PIC/MS-DPC.

9. Dynamic Application Awareness: The packet header is searched to identify the application based on its port, address, or signature. IPS: The packet is searched to identify threats and attacks using the following detection mechanisms.
   • Anomaly—check traffic against protocol standard.
   • Signature—protocol-aware context signature.
   • Backdoor—detect traffic bypassing normal authentication procedures.

10. The application policy request is forwarded to a local policy manager.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

11. The local policy manager compares the identified application against a customer-defined list of application handling instructions.By default, all packets in the session are examined. One user-configurable option is that the session incurs no further analysis. In this case, Dynamic Application Awareness or IPS no longer analyzes this session, and its resources are available to analyze other sessions. Otherwise, the traffic is pushed to the forwarding plane (step 6).

12. **The local policy manager provisions the appropriate enforcement functions on the forwarding plane in real time.**
    - Rate limit traffic, packet drop
    - Classify traffic (DSCP mark for CoS handling)
    - Connection close, block traffic
    - Statistic gathering and logging

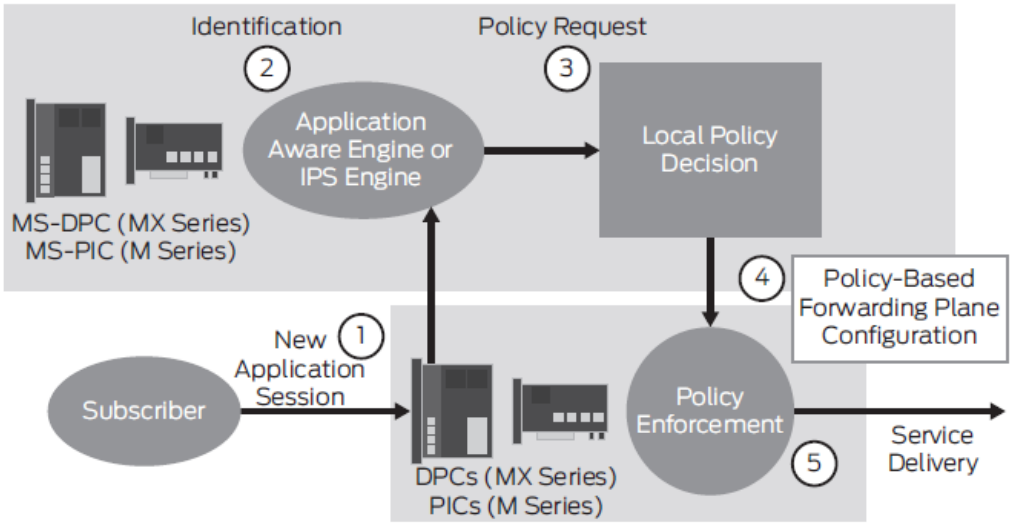


Figure 3: Logical packet flow

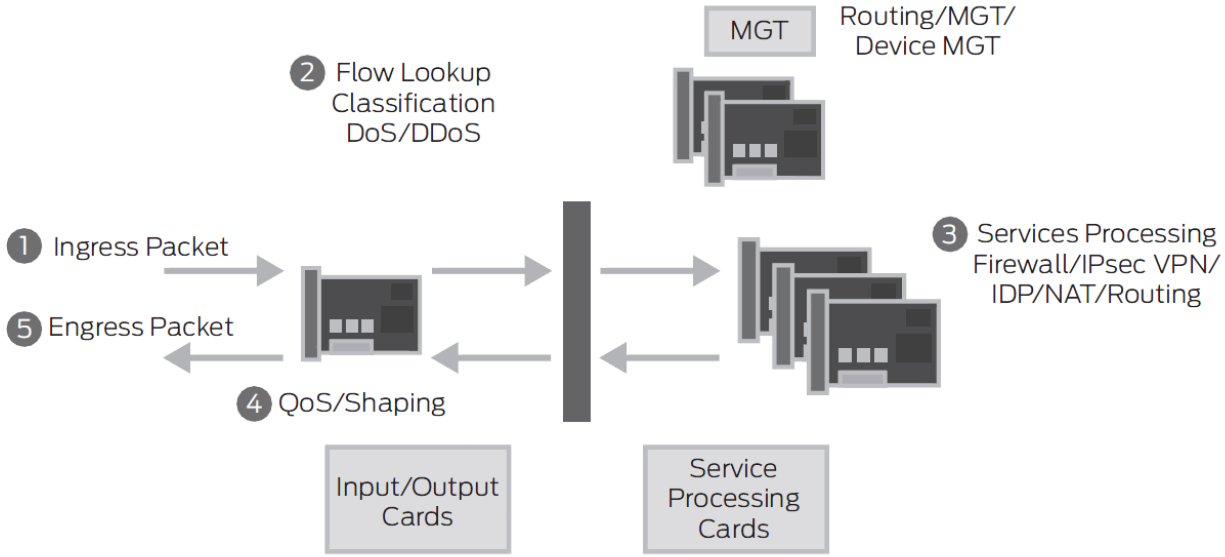
Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Source: *GENERATING NEW REVENUE STREAMS AND INCREASING NETWORK SECURITY Dynamic Application Awareness and Intrusion Prevention System*, Published by Juniper Networks, Dec, 2009, www.**juniper**.net/us/en/local/pdf/whitepapers/2000339-en.pdf **Evidence '163 C1 1c(9)** **Not Just Another Chassis Design** The Dynamic Services Architecture is based on a chassis design; however, it is a complete departure from traditional chassis architecture. Rather than simply providing a fast backplane, the Dynamic Services Architecture includes the management and control necessary to incorporate individual blades into a powerful collective solution. Rather than housing disparate cards, the Dynamic Services Architecture adds each blade into a growing pool of resources. These resources can then be utilized as necessary for optimal processing of network traffic. **Switch Fabric, Control Board and Route Engine** At the heart of the Dynamic Services Architecture is the Switch fabric and Control Board (SCB). The SCB transforms the chassis from a simple blade enclosure into a highly effective mesh network. The purpose of the SCB is to allow all blades in the chassis to send traffic at extremely high bandwidth. The Route Engine (RE) is tightly coupled with the functionality of the SCB and can be considered the central nervous system of the architecture. The RE is the control plane of the chassis and provides overall management and communications to and from system administrators, as well as calculating route tables for routing network traffic. The operating system, which includes key chassis functionality, also runs on the RE. In the case of networking and security, functionalities such as advanced routing, switching, flow-base security, zone-based management, and screens are available on the OS.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Service Processing Cards If the RE is the central nervous system of the chassis, the Service Processing Card (SPC), is the brain. SPCs are blades that provide the capacity to perform the heavy lifting of processing network packets. **Session Distribution** The Dynamic Services Architecture also supports automatic load balancing with advanced performance and capacity due to its session distribution design. This is enabled by the intelligent input/output and network processing subsystems, which balance sessions across the shared pool of SPCs (the "brain" discussed above). This is possible because all the SPCs in the system run the same services and have the same configuration. There is no specific mapping from one IOC to one SPC; rather, each flow is mapped dynamically upon session creation. Any session coming in any port can be forwarded, on a session-by-session basis, to any SPC in the system. This load balancing is performed automatically, with no configuration or oversight by the system administrator. This is dramatically opposed to traditional chassis-based solutions, where each processing blade is an independent firewall, with its own dedicated traffic, unique configuration, and routing support. **Packet Flow** In the same way, the packet flow within the Dynamic Services Architecture becomes fully integrated and far easier to manage. No longer is it necessary for administrators to provide separate instructions to each blade for traffic management. Each packet traversing the system now takes the same basic path: 1. The ingress packet enters Ethernet port on the IOC. 2. It is processed by the IOC and passed to the switch fabric. 3. One processing unit on the SPC receives and processes the packet for the firewall, IPsec VPN, and/or IPS. If the packet is to be dropped, the SPC does so and will typically log the event. 4. If the packet is to be passed, it is passed back through the switch fabric to the IOC, where it is processed by the IOC processor, where QoS is applied if necessary.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	5. The packet is then passed out the Ethernet port to egress the system.  Figure 2: An example of a fully integrated packet flow (SRX5000 line) **Source:** *Dynamic Services Architecture: a Revolutionary Approach to Integrated Network Security*, published by Juniper Networks, Oct 2009 , pages 4-6, https://www.juniper.net/us/en/local/pdf/whitepapers/2000288-en
1d. and storing an indication of each of the identified components so that the nonpredefined sequence does not need to be re-identified for	The accused products store information about the processing components, along with a correlation to the network traffic that those components are to operate on, as defined by the result of the non-predetermined result of the packet classification definitions, the network flows, and the executed policy directives, in accordance with this limitation. **Evidence '163 C1 1d(1)**

subsequent packets of the message;	The security policy to be used for the first packet of a flow is cached in a flow table for use with the same flow and closely related flows. Security policies are associated with zones. A zone is a collection of interfaces that define a security boundary. A packet's incoming zone, as determined by the interface through which it arrived, and its outgoing zone, as determined by the forwarding lookup, together determine which policy is used for packets of the flow. **Source:** *Junos OS Security Configuration Guide,* Published by Juniper Networks, Inc., March 2011, page 4, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf **Evidence '163 C1 1d(2)** Flow-based packet processing, which is stateful, requires the creation of sessions. A session is created for the first packet of a flow for the following purposes: • To store most of the security measures to be applied to the packets of the flow. • To cache information about the state of the flow. For example, logging and counting information for a flow is cached in its session. (Some stateful firewall screens rely on threshold values that pertain to individual sessions or across all sessions.) • To allocate required resources for the flow for features such as NAT. • To provide a framework for features such as ALGs and firewall features Most packet processing occurs in the context of a flow, including: • Management of policies, NAT, zones, and most screens. • Management of ALGs and authentication. **Source:** *Junos OS Security Configuration Guide,* Published by Juniper Networks, Inc., March 2011, page 6, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf]

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

1e. and for each of a plurality of packets of the message in sequence, for each of a plurality of components in the identified non-predefined sequence, retrieving state information relating to performing the processing of the component with the previous packet of the message;	The accused products attempt to retrieve state information for the processing of components every time a packet which qualifies for handling passes through the system, in accordance with this limitation. **Evidence '163 C1 1e(1)** When the application identification feature identifies a new application, it caches the result (the destination address, port, protocol, and service) to reduce processing for subsequent sessions. The application cache and extended application cache are maintained separately. **Source:** *IDP Series Concepts and Examples Guide,* Published by Juniper Networks, Feb. 2011, Page 96, http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf **Evidence '163 C1 1e(2)** The fast-path packet process consists of the following steps: 1. An inbound packet is received by an interface and sent to the NPU, which provides processing for that interface. The NPU performs a session lookup and determines that it knows the session and the SPU processing it. The NPU then forwards the packet directly to the SPU which owns the session. 2. Policing, stateless filtering, and screens are performed. Technically, the screens that are applied after the initial packet setup are all on the NPU on the high-end SRX platforms. 3. The SPU determines if it knows about the session already, which in this case it does. The session entry will provide cached instructions on how to process the packet so that the SRX does not have to do any forwarding or policy checks, as these have already been determined in the first packet processing. **Source:** *Junos Security,* By: Rob Cameron; Brad Woodberg; Patricio Giecco; Timothy Eberhard; James Quin; Publisher: O'Reilly Media, Inc. as part of the Juniper Networks Technical Library, September 7, 2010, ISBN-13: 978-1-4493-8171-4, page 724
1f. performing the processing of the	The accused product performs the processing based on the retrieved state information.

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

identified component with the packet and the retrieved state information;	**Evidence '163 C1 1f(1)** When the IDP engine processes security policy rules, it examines the session, beginning with the first packet, to identify a match. To match service or application, the IDP engine first compares the session against the application identification cache to identify the application. If the session does not match the application identification cache, the IDP engine processes the session against the application signatures. If the IDP engine is still unable to determine the application, it uses the standard application protocol and port. **Source:** *IDP Series Concepts and Examples Guide, Juniper Networks, Published Feb. 2011,* http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, page 96
1g. and storing state information relating to the processing of the component with the packet for use when processing the next packet of the message.	State information for the processing of each of the identified components is stored in an IDP Application System Cache, in accordance with this limitation. **Evidence '163 C1 1g(1)** Application system cache (ASC) saves the mapping between an application type and the corresponding destination IP address, destination port, protocol type, and service. Once an application is identified, its information is saved in the cache so that only one pattern matching is required for an application running on a particular system, thereby expediting the identification process. A mapping is saved in the cache only if the matched signature contains both client-to-server and server-to-client patterns. This process protects the system from hackers who might send malicious packets through a legitimate server port so that it is interpreted as a different application. By default, the application system cache saves the mapping information for 3600 seconds. However, you can configure the cache timeout value. **Source:** *Junos OS Security Configuration Guide,* Published by Juniper Networks, Inc., March 2011, page 802, https://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/junos-security-swconfig-security.pdf

Implicit Networks, Inc.
U.S. Patent No. 6.629,163 C1
Claims Chart
Implicit Networks, Inc. v. Juniper Networks, Inc.
Security (IDP, UTM) Use Case

	Additionally, the accused products can discover their own view of the baseline security state of the network, store this state, and automatically develop security policies to detect and act on behavior which varies from that baseline. **Evidence '163 C1 1g(2)** The Profiler is a network-analysis tool that helps you learn about your internal network so you can create effective security policies and minimize unnecessary log records. The Profiler queries and correlates information from multiple IDP Series devices. After you configure the Profiler, **it automatically learns about your internal network** and the elements that constitute it, including hosts, peers (which host is talking to which other host), ports (non-IP protocols, TCP/UDP ports, RPC programs), and Layer 7 data that uniquely identifies hosts, applications, commands, users, and filenames. You can use this data to investigate and analyze potential problems in the network and to resolve security incidents. During profiling, the IDP Series device records network activity at Layer 3, Layer 4, and Layer 7 and stores this information in a searchable database called the Profiler DB. The Profiler uses session creation, session teardown, and protocol contexts to generate this database, which defines all unique activities occurring on your network. Unique activities include attempts, probes, and successful connections. The device logs normal events only once, and it logs all unique events as often as they occur. Baseline data gives you the building blocks for your network security policy. After you have created a baseline and installed an appropriate security policy, you can use Profiler to alert you when new hosts or applications appear in your network. You can analyze the alerts to decide whether to update your security policy. **Source:** *IDP Series Concepts and Examples Guide, Juniper Networks, Published Feb. 2011,* http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, page 32

EXHIBIT 4

TO BE FILED UNDER SEAL

EXHIBIT 5

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

IMPLICIT NETWORKS, INC.

Plaintiff,

v. Case No. C 10-4234 SI

JUNIPER NETWORKS, INC.

Defendant.

HIGHLY CONFIDENTIAL - UNDER PROTECTIVE ORDER

VIDEOTAPED DEPOSITION OF SCOTT M. NETTLES, Ph.D.

San Francisco, California

October 9, 2012

Reported by:

KENNETH T. BRILL

CSR NO. 12797

Job No. 1538661

PAGES 1 - 285

programs, the expressions of which are not sort of a one-to-one mapping between the machine code and the expression. 09:56:55 09:57:00 09:57:04

And there are a lot of different ways that these higher-level languages can be processed. The typical ways that we talk about it are compilers and interpreters. But, generally, when we're talking about source code, we're talking about, again, assembly language code could be source code, but typically we're talking about programming in these higher-level programming languages. And the code that's written in those higher-level programming languages is called the source code, and that's because it's the source of the instructions for the computer even though there's going to be a translation that's perhaps nontrivial down into the machine code. 09:57:07 09:57:09 09:57:13 09:57:17 09:57:20 09:57:26 09:57:29 09:57:31 09:57:36 09:57:39 09:57:41 09:57:43 09:57:46 09:57:49

So that's -- that's basically what source code is. 09:57:50 09:57:53

Q. So is it fair to say that source code is the human-readable instructions that describe how software works? 09:58:02 09:58:05 09:58:13

A. I think it would be more to say it's the human-writable expression that describes how -- how -- the instructions that you're giving to the 09:58:24 09:58:26 09:58:30

computer. But it -- when something is human-writable, it's typically human-readable as well. 09:58:32 09:58:37 09:58:40

Q. Maybe we could say that source code is human-understandable instructions that describe how software works, is that fair? 09:58:43 09:58:45 09:58:48

A. Well, again, I -- I made a deliberate distinction between readability and writability because one of the kind of interesting things that seems to be a deep truth about computer science is that understanding something is simpler than producing something. 09:58:51 09:58:56 09:58:58 09:59:03 09:59:06 09:59:12

So we have lots of different kinds of algorithms that can very easily understand something but can't necessarily very easily produce something. 09:59:13 09:59:15 09:59:20

But yes, source code is the -- is the way that computer programs read and write their algorithmic and other programming ideas both to the computer and to each other. 09:59:24 09:59:27 09:59:31 09:59:37

Q. Source code is a way that humans give instruction to computers -- 09:59:40 09:59:45

A. And also -- 09:59:49

Q. -- correct? 09:59:50

A. Yes, and also, you know, show each other how -- how those -- what those instructions are. 09:59:51 09:59:54

Q. And -- and that's right, programmers sometimes put in comments into the source code which are ignored by the computer but are helpful to other humans who may want to understand how the source code operates; correct? 09:59:57 10:00:01 10:00:05 10:00:09 10:00:12

MR. HOSIE: Objection, vague and ambiguous, overbroad. 10:00:13 10:00:14

THE WITNESS: Yes, although it's -- it's another one of these places where there's a term that's kind of used in a general way and a term that's used in a -- in a specific way. 10:00:16 10:00:22 10:00:24 10:00:27

So I think we might often talk about source code referring to everything that's written, but I think if you ask a programmer, typically they'd say, well, no, the comments really aren't the source code, the source code is just the literal expressions that turn into machine instructions. 10:00:29 10:00:32 10:00:35 10:00:38 10:00:42 10:00:45

But yes, there are certainly comments in -- well, there are not as many comments as you would like to have in source code, but you can put them there and sometimes they're -- they're apparent. 10:00:47 10:00:49 10:00:52 10:00:55 10:00:59

BY MR. McPHIE: 10:01:00

Q. Would you agree that source code is the best evidence of what software actually does? 10:01:00 10:01:02

fair to divide by four, but I wouldn't -- you know, that -- that -- that's definitely a guesstimate. 11:27:52 11:27:56

BY MR. McPHIE: 11:28:03

Q. So maybe 75 to 100 hours total for the Juniper infringement analysis? 11:28:03 11:28:07

MR. HOSIE: Objection, vague and ambiguous. 11:28:09 11:28:10

THE WITNESS: Yeah, I mean, I -- I wouldn't -- I wouldn't want to be held to either that minimum or maximum, and I don't think I could -- it would -- it would be difficult for me to -- to break it all out. 11:28:10 11:28:12 11:28:19 11:28:22 11:28:24

I mean, part of it is just because a lot of time is spent understanding the claims and the -- and the specification of the Claim Construction Order, and that applies to everybody sort of equally, so... 11:28:26 11:28:29 11:28:31 11:28:33 11:28:36

BY MR. McPHIE: 11:28:39

Q. One aspect of the infringement analysis is that unlike the validity analysis, there is little or no overlap between the work that you do for the F5 Networks case and the work you do for the Juniper Networks case, is that fair? 11:28:40 11:28:43 11:28:47 11:28:51 11:28:54

A. Well, again, there's, I think, substantial overlap in that, you know, there is -- there is the 11:28:58 11:29:00

understanding of the patents, there is the understanding of the claim construction, that plays a pretty crucial role. 11:29:03 11:29:05 11:29:10

In terms of understanding the -- the specific evidence and the specific way the systems work, there still is some overlap because the systems are similar, and so when you understand better how one of them works, that helps you understand the other one and -- and vice versa. 11:29:12 11:29:14 11:29:16 11:29:25 11:29:28 11:29:30

But yes, there's substantially less overlap. And in the case of the invalidity rebuttal, there is a substantial amount of overlap, I think that would be a fair way of -- it would be a more precise way of -- of making that distinction. 11:29:33 11:29:37 11:29:40 11:29:42 11:29:45

Q. For example, it would not be proper to rely on the F5 Networks source code in support of your infringement opinions for the Juniper accused products; correct? 11:29:51 11:29:53 11:29:59 11:30:04

MR. HOSIE: Objection, vague -- 11:30:05

THE WITNESS: That's correct. 11:30:07

MR. HOSIE: -- and ambiguous. 11:30:08

BY MR. McPHIE: 11:30:20

Q. Now, for the work that you did reviewing source code -- let me withdraw that. 11:30:20 11:30:23

You did, in fact, review the Juniper 11:30:25

source code in connection with your infringement analysis in this case; correct? 11:30:27 11:30:30

A. That's correct. 11:30:32

Q. And you were aware that Juniper made source code available on a secured computer that was available in the offices of counsel; correct? 11:30:35 11:30:39 11:30:45

A. Yes, sir. 11:30:48

Q. You yourself did not actually come to the attorney's offices and sit down at the source code computer directly; correct? 11:30:50 11:30:55 11:31:00

A. That's right. I had an assistant who did that. 11:31:02 11:31:05

Q. Who was that assistant? 11:31:05

A. His name is Pavel, but it's a Russian last name, and I'm afraid my -- my bad memory for names is -- is escaping me. 11:31:07 11:31:13 11:31:18

Q. We'll call him Pavel. 11:31:20

A. Okay. 11:31:23

Q. Do you know, is that P-A-V-A-L? 11:31:24

A. I think it's E-L. 11:31:26

Q. E-L, all right. 11:31:28

Is Pavel someone -- you said he is your assistant? 11:31:29 11:31:33

A. Well, that's the role he played here. 11:31:34

Q. Okay. I should ask, in your role as an 11:31:36

expert witness, is that done under the auspices of a -- of some sort of company or is it just Scott Nettles? 11:31:42 11:31:45 11:31:49

A. It's just Scott Nettles. 11:31:50

Q. Okay. Are there others that you -- that are affiliated with or associated with you who -- who come along and do projects for you or with you as you do these expert engagements? 11:31:52 11:31:54 11:31:58 11:32:01

A. Well, generally, the law firms that I work with engage the additional assistants. There are some people who I've worked with multiple times, but they're always engaged independent of me. They're not engaged through me. 11:32:10 11:32:13 11:32:20 11:32:25 11:32:28

I guess there have been a few times a long time ago where I -- I did engage someone to act as an assistant directly, but that hasn't been my practice for a long time. I can only think of one instance of that, actually. So -- but there are some people I work with on a recurring basis. 11:32:30 11:32:34 11:32:41 11:32:45 11:32:51 11:32:54

Q. Okay. In other words, it was the offices of Hosie Rice that hired Pavel and not you, is that correct? 11:32:57 11:33:00 11:33:04

A. That's correct. 11:33:05

Q. Okay. Had you ever met Pavel prior to working with him on this matter? 11:33:06 11:33:10

Q. And was it your understanding that Pavel had in fact already done some source code review at the time that you started working on the Implicit matters? 11:34:37 11:34:38 11:34:43 11:34:46

A. Yes, sir, that's my understanding. 11:34:46

Q. And specifically, he had spent some time reviewing the Juniper source code? 11:34:48 11:34:51

A. Yes, although I'm not really aware of exactly when he -- I don't -- I don't know what the overlap or -- or lack of overlap is. 11:34:54 11:34:56 11:35:01

Q. Do you know how long Pavel spent reviewing the Juniper source code? 11:35:10 11:35:13

A. Not precisely. I would assume that you would know. 11:35:15 11:35:17

Q. Does the amount of time that he spent reviewing source code, the source code in this case, would that factor into your analysis at all? 11:35:21 11:35:23 11:35:28

A. No. In source -- you know, throughout this whole discussion this morning, you've sort of acted like source code analysis is some kind of a -- a gold standard and, you know, the only kind of evidence that you could -- you could generate. 11:35:37 11:35:43 11:35:46 11:35:50 11:35:53

But actually, my experience in doing quite a bit of source code analysis is that you often spend a lot of time looking at things that are not 11:35:55 11:35:58 11:36:02

important, and because it's a static analysis, you may not be getting -- be getting a clear picture of exactly what's -- what's happening in the product. And so the amount of time that you spend, it may reflect more the complexity of the source code tree than really the -- the details of your analysis. 11:36:06 11:36:12 11:36:15 11:36:20 11:36:26 11:36:30

I mean, I've done analyses of source code where the amount of source code was very small, so it doesn't take very long to do a thorough review. And I've been involved in reviewing source code that's very large, and there the length of time that it takes often is a function of how long it takes to narrow yourself down to the source code you want to look at carefully. So I don't think -- I don't think the amount of time would really bear in to that at all in a clear way. 11:36:33 11:36:36 11:36:38 11:36:42 11:36:45 11:36:47 11:36:52 11:36:55 11:36:57 11:37:01

Q. Did you ever ask Pavel how much time he spent reviewing the source code? 11:37:03 11:37:06

A. No, sir, I did not. 11:37:07

Q. Did you ever ask Pavel what was his experience reviewing or working with the C programming language? 11:37:12 11:37:14 11:37:18

A. No, sir, I did not. 11:37:20

Q. Is it your understanding that the Juniper source code is written in C? 11:37:26 11:37:28

A. Yes, sir, it is. 11:37:30

Q. Now, to what extent -- well, withdrawn. 11:37:38

Was there any interactivity between you and Pavel in terms of how he would go about doing his source code review on the computer at counsel's office? 11:37:45 11:37:48 11:37:51 11:37:54

A. No, sir, there was not. 11:37:54

Q. What was the work product that you received from Pavel as a result of his source code review? 11:37:57 11:38:01 11:38:05

A. So I received -- so as you noted, I didn't actually go and look at the source code on the source code machine, but Pavel printed certain portions of the source code with Bates numbers and so on and so forth. 11:38:14 11:38:16 11:38:19 11:38:24 11:38:28

And so I received a copy of that, and then I received various analyses that he did of that source code. And I received versions of the infringement contentions that reflected his analysis. 11:38:29 11:38:32 11:38:37 11:38:42 11:38:46

Q. You understood that you were welcome and invited to come to our offices to review the source code, the Juniper source code on the secured machine in our -- in our offices; correct? 11:38:49 11:38:52 11:38:55 11:38:58

MR. HOSIE: Objection, vague and 11:39:03

ambiguous. 11:39:03

THE WITNESS: I -- I don't really think I was quite aware of what the current status of source code discovery was at the time I was working primarily on this case. So I -- I don't know for a fact one way or the other. 11:39:13 11:39:16 11:39:21 11:39:24 11:39:27

BY MR. McPHIE: 11:39:28

Q. No one ever told you that you were free to come to the offices of counsel to personally conduct a review of the Juniper source code? 11:39:29 11:39:31 11:39:35

MR. HOSIE: Objection. Under the federal rules, conversations between counsel and the expert are not the proper subject of discovery. If you'd like to rephrase that for conversations ex counsel. 11:39:38 11:39:42 11:39:45 11:39:47

BY MR. McPHIE: 11:39:54

Q. You had no understanding one way or the other as to whether you were free to come to the offices of counsel to personally conduct a review of the Juniper source code, is that right? 11:39:54 11:39:57 11:40:02 11:40:05

A. I think answering that question would reveal communications that I've had with counsel about the preparation of my report. 11:40:08 11:40:10 11:40:12

Q. Well, I'm just asking what was your understanding. 11:40:15 11:40:17

A. But I think telling you my understanding 11:40:23

the SRX. But I'm not -- I'm not certain about that either. So, for example, on page 27 we see a picture. 01:23:52 01:23:56 01:24:02

See, that's not a reference to flowd. You -- I'd either need to refer to specifically to references to flowd in the report or you'd need to direct me to one. 01:24:08 01:24:10 01:24:15 01:24:18

Q. When you say the services routers, what are you referring to? 01:24:20 01:24:22

A. Well, my understanding is that initially there were some additional products that were accused, and those routers used -- and I apologize, I don't remember all the details of that -- and it's -- it's not in my report because they were -- they were dropped. 01:24:23 01:24:28 01:24:31 01:24:36 01:24:37 01:24:40

Those routers required an additional board, or component, maybe it wasn't a board, called something like a services PIC to do the same kind of flow-based routing are -- that is built into and is a fundamental part of the J series and the SRX series. 01:24:44 01:24:48 01:24:51 01:24:57 01:24:59 01:25:03

And so when I say the services routers, I was thinking about those. I don't -- I don't have a specific set of model numbers at the -- at my fingertips. 01:25:03 01:25:06 01:25:08 01:25:11

Q. If I were to mention the M-series, MXT, TX series with the multi-services PIC or adaptive services PIC, does that ring a bell in terms of what was included in that category of formerly accused Juniper products? 01:25:13 01:25:18 01:25:23 01:25:26 01:25:29

A. It does. What I don't remember is of those various families, were all of them capable of using the multi-services PIC or not. 01:25:33 01:25:36 01:25:40

And as I remember there were a couple of different flavors of that -- of that board or enhancement, whatever it was, and I think that's only one of the names. So I don't really remember the details of exactly how it -- how it works. But those model numbers -- or those model designations, I guess they're not specific numbers, sound familiar. 01:25:43 01:25:46 01:25:49 01:25:53 01:25:55 01:25:58 01:26:02 01:26:04

Q. Okay. But we could call those the services routers, that's how you understood them? 01:26:05 01:26:07

A. That was the shorthand I used a few minutes ago. I think that's fine. 01:26:10 01:26:11

Q. Did you do any analysis with respect to the services routers? 01:26:13 01:26:15

A. I didn't do any analysis that I eventually relied on in the report. 01:26:30 01:26:33

Q. Did you ever reach an opinion that the 01:26:35

services routers did not infringe the patents in suit? 01:26:37 01:26:40

A. I don't think I'm at liberty to answer that question. 01:26:56 01:26:57

Q. Why not? 01:26:58

A. Well, I think that that falls under the -- the category of report preparation, and since I didn't eventually rely on that opinion in my -- in my report, I don't -- I don't think that -- my understanding is that opinion is not discoverable. 01:27:02 01:27:07 01:27:12 01:27:15 01:27:19

MR. McPHIE: Mr. Hosie, do you share the view that Dr. Nettles has just expressed? 01:27:24 01:27:25

MR. HOSIE: I will think about that and talk to you off the record. What I will not do is debate it on the record now because it's inappropriate. 01:27:33 01:27:36 01:27:43 01:27:47

MR. McPHIE: Are you -- are you going to basically issue an instruction not to answer? 01:27:52 01:27:53

MR. HOSIE: Let me take a break. I'll talk to the witness. 01:27:59 01:28:00

MR. McPHIE: Well, it's just a legal question. I mean, are you going to give him an instruction not to answer or not? 01:28:02 01:28:03 01:28:06

MR. HOSIE: Are you objecting to our taking a break? 01:28:08 01:28:09

MR. McPHIE: Well, I -- I tend not to not like to have breaks when there is a question pending. So... 01:28:10 01:28:12 01:28:15

MR. HOSIE: Well, he -- he answered the question, then you asked me a question and I'm not the deponent here. So there isn't a question pending and so we're going to take a break. 01:28:16 01:28:18 01:28:20 01:28:23

MR. McPHIE: Well, he refused to answer the question. He said, I'm not at liberty to say, and I'm asking, you know, for whether that's your view as well. 01:28:25 01:28:27 01:28:29 01:28:32

MR. HOSIE: Right. 01:28:33

MR. McPHIE: If not, I'm going to ask the question again, and we'll get an answer, which is my preference, rather than having him step out for a conference with counsel. 01:28:33 01:28:36 01:28:39 01:28:40

MR. HOSIE: Why don't we take a break. 01:28:41

MR. McPHIE: Well, I'd like not to. 01:28:43

MR. HOSIE: Well, we'll be shortly back. You can stay on the record, if you'd like. 01:28:46 01:28:48

MR. McPHIE: I'll note for the record that counsel and the witness have left the room. 01:28:53 01:28:54

THE VIDEOGRAPHER: Do you want to stay on the record? 01:29:07 01:29:08

MR. McPHIE: Actually, let's turn it off, 01:29:10

because there's no reason to burn the time. 01:29:12

THE VIDEOGRAPHER: Off the record at 01:29:15

1:28 p.m. 01:29:16

(Recess taken.) 01:29:58

THE VIDEOGRAPHER: Back on the record at 01:29:58

1:29 p.m. 01:30:00

BY MR. McPHIE: 01:30:00

Q. Dr. Nettles, did you form an opinion that 01:30:01

the services routers from Juniper did not, in fact, 01:30:04

infringe the patents in suit? 01:30:10

MR. HOSIE: If I could -- excuse me, if I 01:30:13

may, the question asks if you formed an opinion. It 01:30:14

does not ask for the content of conversations with 01:30:18

counsel, so I think it's a proper question. 01:30:21

THE WITNESS: Oh, so I neither formed an 01:30:24

opinion that they did infringe nor that they did not 01:30:27

infringe. 01:30:31

BY MR. McPHIE: 01:30:33

Q. Did you perform any analysis regarding the 01:30:34

Juniper services routers with respect to 01:30:39

infringement? 01:30:41

A. Yes. And the result of that analysis was 01:30:47

that I didn't actually get to the point that I 01:30:50

formed an opinion about whether or not they 01:30:54

infringed or not. 01:30:56

Q. Earlier I believe you may have testified something regarding flowd and its relationship with the services routers. 01:31:06 01:31:12 01:31:15

A. Well, that was -- that was the reason I was looking for specific references to flowd, because it -- it may be that the -- the specific references that I saw to flowd were in that context. And I -- and I -- and I can't remember, and I haven't looked through my report carefully to see if there are any references here nor, you know, have I looked at the extensive documents that are cited in the report to see if there's a reference to flowd. 01:31:19 01:31:21 01:31:25 01:31:36 01:31:40 01:31:42 01:31:45 01:31:49 01:31:51

I remember a diagram that had a label flowd. And I think that -- that diagram may have been a services router diagram rather than an SRX or J series diagram, but I -- I'm not sure about that. 01:31:54 01:31:57 01:32:01 01:32:06

Q. In your view why does it matter whether the facts under consideration in this case pertained to one of the accused products or one of the Juniper services routers? 01:32:14 01:32:16 01:32:19 01:32:24

MR. HOSIE: If I could have that read back, please. 01:32:25 01:32:26

- - - 01:32:36

(The court reporter read back as follows: 01:32:36 01:32:36

"QUESTION: In your view why does it matter whether the facts under consideration in this case pertained to one of the accused products or one of the Juniper services routers?") 01:32:36

- - - 01:32:36

THE WITNESS: Well, at -- at some level it doesn't. I mean, Juniper has basically represented that their products all work the same. But if it's about the services routers, then there's a good chance that I didn't include that picture in my -- in my report, so there really would be no reason to look hard for that -- that picture. 01:32:39 01:32:41 01:32:44 01:32:49 01:32:54 01:32:57 01:33:00

And I think that the -- the evidence that I've cited here, which is clearly directed at the SRX and J series, is substantial, so I'm concerned about whether or not that picture appears or not. 01:33:03 01:33:08 01:33:11 01:33:19

BY MR. McPHIE: 01:33:24

Q. Can you point me to any information regarding the operation of flowd that you do rely upon in your infringement report? 01:33:25 01:33:26 01:33:30

A. Well, for example, I cite to depositions by Mr. Krishna and Mr. Tavokoli. I think those depositions may talk about flowd. 01:35:03 01:35:09 01:35:14

Q. Did you cite to the portions of the 01:35:19

deposition where flowd was discussed, or do you just not remember sitting here today? 01:35:22 01:35:27

A. I don't remember that specifically, I'm still looking for specific references. 01:35:29 01:35:31

Q. Yes. Please, if you can, point me to any other material in your report that relies on the operation of flowd, please do so. 01:35:32 01:35:35 01:35:38

A. I mean, it's important to understand that the report is a disclosure of the evidence that I expect to use. And so, in many cases the report is making references to other sources of evidence, in particular, for example, the Enterprise Routing Book has a great deal of information about how these systems work as well as the security -- I forgot exactly what it's called, but the security book. And there certainly could be references there that wouldn't have been called out explicitly in the report. I guess the securities book is called Junos Security. 01:35:41 01:35:43 01:35:46 01:35:49 01:35:54 01:35:59 01:36:02 01:36:07 01:36:09 01:36:12 01:36:14 01:36:40

Here's a citation to a work called Dynamic Surfaces Architect, that's essentially a place that I would go to look further for -- 01:37:24 01:37:29 01:37:33

Q. Do you have a page number for that? 01:37:37

A. Sorry. On page 49, I think it's cited in the report in other places as well. 01:37:39 01:37:42

MR. HOSIE: Objection, asked and answered. 01:56:40

THE WITNESS: I think I answered that, but 01:56:43

no, I -- I did not ask if I could -- Pavel if I 01:56:44

could look at the whole source code tree because if 01:56:48

I had asked that question, I would have answered yes 01:56:51

to the previous question involving this specific a 01:56:53

release. 01:56:58

BY MR. McPHIE: 01:56:59

Q. What is Profiler? 01:57:00

MR. HOSIE: Objection. Vague and 01:57:04

ambiguous. 01:57:05

THE WITNESS: I -- I don't -- you have to 01:57:06

give me more context than that. 01:57:08

BY MR. McPHIE: 01:57:10

Q. Was Profiler something that you rely upon 01:57:10

in support of your infringement opinions in this 01:57:14

case? 01:57:16

A. So I've looked for a reference to Profiler 01:59:39

briefly in my report and I didn't find such a -- a 01:59:44

reference, but it's important to understand that in 01:59:51

this particular case the primary evidence that I 01:59:53

relied upon isn't the code citations that we've been 01:59:59

focusing on. It's really the citations to 02:00:04

deposition testimony and especially to the extensive 02:00:06

Juniper documentation about how the system 02:00:10

functions. So including this entirety security -- 02:00:12
Junos Security Book, which is a very detailed 02:00:16
explanation of how the system works. 02:00:21
So perhaps there is a reference to 02:00:23
Profiler in here, but I -- I can't find it in any 02:00:26
quick manner and I don't specifically remember it. 02:00:32
Q. It is the non-source code evidence that 02:00:38
you are most familiar with, is that fair? 02:00:42
A. I think it's the non-source code evidence, 02:00:48
especially the extensive documentation, the diagrams 02:00:50
we've been talking about, which provides the 02:00:54
broadest and strongest support for infringement in 02:00:57
this case. 02:01:00
Q. And it's also that non-source code 02:01:07
evidence that you are the most comfortable and -- 02:01:09
and familiar with? 02:01:11
A. Well -- 02:01:14
Q. Is that right? 02:01:16
A. Source -- that's right, but it -- it 02:01:17
deserves an explanation. Source code is very 02:01:22
detailed. It's not something that you're going to 02:01:24
recollect the details of in a deposition where 02:01:27
you're looking at it on the fly and you know that 02:01:33
you have a limited amount of time to answer these 02:01:38
sorts of -- these sorts of questions. 02:01:41

So, but yes, I think that I focused on the evidence that I thought was the strongest support and that was documentation, deposition testimony, more so than source code, and I -- I am more familiar with that. But again, that would be expected just because of the nature of source code. 02:01:44 02:01:46 02:01:49 02:01:54 02:01:57 02:01:59

Q. And your opinion is that the strongest source of evidence supporting your opinions regarding infringement in this case are the non-source code pieces of evidence; correct? 02:02:03 02:02:05 02:02:09 02:02:15

MR. HOSIE: Objection, vague and ambiguous. 02:02:18 02:02:19

THE WITNESS: Well, again, there is a lot of evidence about how the system operates, and that's evidence that Juniper has published, makes available to their customers, that their customers and their customers' employees rely upon, and I think it's very strong evidence of infringement. 02:02:23 02:02:25 02:02:29 02:02:33 02:02:36 02:02:40

Source code, although it can be a good source of -- of evidence, is a static analysis of a complicated system. It's not always clear that the right things have been analyzed. It's not always clear that the source code that happens to be referred to is really the -- the most important source code. And so in this case I have relied more 02:02:44 02:02:48 02:02:53 02:02:56 02:02:59 02:03:01 02:03:09

heavily on documentation. And I think that's appropriate. 02:03:12 02:03:15

BY MR. McPHIE: 02:03:15

Q. When you say that source code analysis is a static analysis, what do you mean? 02:03:18 02:03:22

A. Well, it -- it means that the analysis of source code as it's been conducted here. And to the best of my knowledge, this is the only option that was available, looks at the source code as a -- a static object, not as an actual executing entity. 02:03:35 02:03:37 02:03:44 02:03:46 02:03:49

And in computer science, that's important because we know that if you do a static investigation of the properties of a program, that there are many properties of a program that are important that you cannot establish that way. And you can only establish certain properties of a program by a dynamic analysis by running the program. 02:03:53 02:03:57 02:04:00 02:04:03 02:04:05 02:04:08 02:04:11 02:04:16

And so what we haven't had the opportunity to do here is to run this program, observe its behavior using, say, a program debugger, so that we could really see what happens when we run -- when we execute things. 02:04:16 02:04:19 02:04:22 02:04:25 02:04:28

So that's a real limitation of source code analysis is that it's -- especially if it's not 02:04:29 02:04:31

accompanied by other kinds of source-related analysis, is that it's not observing a running system, and we know as a fact that there are certain key aspects of a computer program that simply can't be established through static analysis. 02:04:35 02:04:38 02:04:42 02:04:47 02:04:51

Q. You're saying that there are aspects of computer software that cannot be established by reviewing the source code, or just that it is difficult to understand those aspects? 02:04:57 02:04:59 02:05:01 02:05:05

A. It is mathematically impossible to establish certain aspects of computer programs through static analysis and inspection of the code. It's a very deep and well understood idea in computer science and the things that can't be established are actually very fundamental and very basic. 02:05:12 02:05:14 02:05:18 02:05:21 02:05:23 02:05:25 02:05:28

Q. Are there any aspects of the claims in -- withdrawn. 02:05:29 02:05:34

Are there any aspects of the asserted patent claims in this case that are impossible to determine by inspection of source code? 02:05:34 02:05:37 02:05:40

A. That's an interesting question. I haven't been asked to do that analysis. I'd be hesitant to try to do that analysis on the fly because it involves doing mathematical proofs of a form that 02:05:47 02:05:51 02:05:59 02:06:02

are notoriously tricky. There are clearly aspects of the claims that would be more incontrovert- -- more incontrovertibly established by observing a running system, especially with a debugger than by static analysis of looking at source code. That -- that's certainly true. 02:06:06 02:06:18 02:06:24 02:06:28 02:06:32 02:06:35

Q. Did you ask Pavel if he had access to a debugger or any other software tools to understand the source code? 02:06:37 02:06:41 02:06:44

A. I didn't ask Pavel that and I'm not at liberty to explain some of my understanding of that issue further. 02:06:51 02:06:54 02:06:57

MR. HOSIE: So if I may, as I believe your answer indicates, please do not reveal the content of conversations with counsel relating to Juniper's refusing to provide source inspection tools. 02:07:01 02:07:04 02:07:09 02:07:12

THE WITNESS: Yes, sir, I think I avoided making that -- such a disclosure. 02:07:18 02:07:21

MR. HOSIE: Thank you. 02:07:23

MR. McPHIE: And I think we may disagree on that point, but we have to change the tape, so let's go off for a moment. 02:07:23 02:07:25 02:07:27

THE VIDEOGRAPHER: This ends media No. 2 in the deposition of Dr. Scott Nettles. Off the record at 2:07 p.m. 02:07:29 02:07:31 02:07:33

(Recess taken.) 02:18:49

THE VIDEOGRAPHER: Back on the record at 02:18:50
2:18 p.m. This is the beginning of media No. 3 in 02:18:51
the deposition of Scott Nettles. 02:18:54
BY MR. McPHIE: 02:18:57
Q. What is application identification in the 02:18:58
context of the accused Juniper products? 02:19:00
MR. HOSIE: Thank you. 02:19:18
THE WITNESS: I don't -- I don't -- I 02:19:26
don't remember specifically discussing that in my 02:19:28
report. I -- if -- I -- I can look, but I don't -- 02:19:30
I don't remember. 02:19:33
BY MR. McPHIE: 02:19:36
Q. And even separate and apart from what is 02:19:37
in your report, do you have any understanding, 02:19:39
sitting here right now, of what application 02:19:40
identification is in the Juniper accused products? 02:19:46
A. No, I mean, I don't -- unless it's cited 02:20:10
in my report, I don't -- I don't remember and I 02:20:12
don't remember that being a -- particularly 02:20:16
important to my analysis, but there is a lot of 02:20:20
documentation and code so I may not remember without 02:20:23
looking at my report. 02:20:27
Q. And in general your opinion is that the 02:20:31
non-source code sources of supporting evidence are 02:20:35

more reliable than the source code evidence; correct? 02:20:41 02:20:46

A. Well, I think they're broader. I think they represent Juniper's understanding of an actual running system as opposed to a static analysis. There are some questions in Dr. Alexander's report about the applicability of the source code analysis to the specific accused products. I don't necessarily agree with his analysis, and his analysis is very limited in a lot of different ways so that makes me hesitant to agree with him. 02:20:48 02:20:50 02:20:54 02:20:56 02:21:05 02:21:06 02:21:11 02:21:15 02:21:18

I have no question that the Junos Security Book is hundreds of pages of evidence concerning the function and operation of the SRX. I have no question that there is documents that make it clear that the J series and the SRX series operate in the same manner with respect to the infringing functionality. 02:21:22 02:21:27 02:21:32 02:21:36 02:21:39 02:21:44 02:21:47

I have no question that there is extensive documentation in the Enterprise Routing Book about the function of these systems. So I think that -- and -- and in addition there is deposition testimony which talks about how the systems work, talk about specific claim elements and steps, and so -- and to me that's stronger -- I mean, it's not that the code 02:21:47 02:21:49 02:21:52 02:21:58 02:22:00 02:22:04 02:22:10

of is this piece of evidence better or is this piece of evidence worse? It's really more about, well, here's some evidence and here's some additional evidence and here's some additional evidence. 02:26:47 02:26:50 02:26:53 02:26:56

Q. Okay. 02:26:58

A. And so the kind of critique that you're talking about would seem wholly out of place in -- in my report. 02:26:59 02:27:01 02:27:06

Q. And, in fact, you do not believe it is there; correct? 02:27:08 02:27:10

A. I don't recall a place in my report where I said this evidence is better kinds of evidence than others because again, that would -- that's not the analysis I'm doing. It really doesn't have any -- you know, I understand that you're trying to attack my analysis and so you want to make that sort of -- sort of comparison, but it's not what I was attempting to do here. And so if I had such a comment, I would be surprised. 02:27:15 02:27:18 02:27:21 02:27:23 02:27:27 02:27:29 02:27:32 02:27:35 02:27:37

I think the comment that I just read into the record makes it clear that I think that there is some evidence which is very good evidence and, you know, whether or not it's better evidence or not, I -- I haven't opined about in the report. 02:27:39 02:27:41 02:27:44 02:27:48 02:27:50

Q. And your opinion -- well, withdrawn. 02:27:53

And your report doesn't contain any qualification caveat or caution against overreliance on source code, is that fair? 02:27:55 02:27:59 02:28:05

A. I don't think that my report has overly relied on source code. I think it's used source code as support for infringement. I think it's taking Juniper's statements about the applicability of the source code at, you know, based on Juniper's representations in a way that perhaps is -- is -- has been called into question, but that's your problem. So I don't think there's an overreliance, why would I put a caveat in that I don't believe? 02:28:08 02:28:09 02:28:13 02:28:16 02:28:19 02:28:25 02:28:29 02:28:33 02:28:38

Q. And that's all my question is. Your report contains no caveat or caution against overreliance on source code, mainly because you didn't think it was necessary; correct? 02:28:41 02:28:43 02:28:47 02:28:51

A. Mainly because I don't think I overly relied on source code. I think that I rely on lots of different kinds of evidence and that source code is one of the pieces of evidence that I rely upon. 02:28:53 02:28:56 02:28:59 02:29:02

Q. Right. And so there is no caveat or caution against reliance on source code in your report; correct? 02:29:05 02:29:07 02:29:09

MR. HOSIE: Objection, asked and answered. 02:29:10

THE WITNESS: Well, again, you know, I 02:29:11

don't think such a caveat would have been appropriate because I haven't overly relied on source code. I have relied on lots of different sources of evidence. It's really in your analysis by your expert where his focus has been almost entirely on deficiencies in the source code, almost wholly ignoring the other evidence in the report. 02:29:13 02:29:15 02:29:17 02:29:21 02:29:24 02:29:29 02:29:31

There's -- there's where I think there's a real deficiency in the analysis in terms of source code, because he has overly emphasized source code. 02:29:35 02:29:38 02:29:40

I think that I have presented the source code that I have and that it supports my understanding and I think that the other evidence supports it. 02:29:44 02:29:45 02:29:47 02:29:51

BY MR. McPHIE: 02:29:53

Q. And I just want to be very specific here on this point. You said that such a caveat would not have been appropriate. I want to confirm, and in fact such a caveat does not exist; it is not within the pages of your report and the accompanying material; correct? 02:29:54 02:29:56 02:29:59 02:30:04 02:30:08 02:30:11

MR. HOSIE: Objection, asked and answered. 02:30:12

THE WITNESS: Again, I don't think that my report overly relies on source code. I think it uses source code as one part of the evidence. I 02:30:14 02:30:16 02:30:19

don't think it would have been -- been appropriate 02:30:23
to put such a caveat into my report because it would 02:30:25
not have been true. And since I believe my report 02:30:28
is true, there isn't such a caveat. 02:30:33
BY MR. McPHIE: 02:30:39
Q. Okay. We went on a detour -- 02:30:39
MR. HOSIE: I think it was actually a 02:30:55
frolic. 02:30:56
BY MR. McPHIE: 02:30:56
Q. We went on either a frolic or a detour, 02:30:57
but I would like to come back from it, whatever it 02:31:00
was, if that's okay. 02:31:02
MR. HOSIE: By all means. 02:31:04
THE WITNESS: I don't know where you're 02:31:05
going, so I don't know if it was a detour or -- or 02:31:06
the freeway. 02:31:09
BY MR. McPHIE: 02:31:10
Q. Well, let's -- let's go to Exhibit 208 02:31:10
where we have the broken-down elements of Claim 1 of 02:31:12
the '163 patent. 02:31:16
A. Okay. 02:31:18
Q. And I had asked a question and I'll ask it 02:31:22
again now. Could you point me to the portion of 02:31:25
your report where you provide the factual basis for 02:31:30
your opinion that Juniper satisfies element 1g? 02:31:34

A. Okay. So I'll -- I'll begin by saying that I'm looking in my appendix and the reason is that that's where most of the element support is, but there certainly could be further support in the main document and in particular in the depositions that are -- are referenced in the main document about this point. 02:31:40 02:31:54 02:31:58 02:32:00 02:32:03 02:32:06 02:32:10

So I -- I don't want to suggest that the main document is devoid of -- of such evidence. The first -- the first obvious place that I see -- and there may have been an earlier one, but this is the first one that I see, is on page 8 and there is a picture of -- that we've seen, talked about several times, where there is a match session and then there's a -- a yes and a -- well, like I said, there's only a yes in this picture. 02:32:11 02:32:14 02:32:23 02:32:32 02:32:35 02:32:38 02:32:45 02:32:47 02:32:50

And there are a series of processing steps, both in the first path and in the fast path. And a number of those processing steps are going to manipulate state and therefore are going to read state, process state, and write state. 02:32:52 02:32:55 02:33:01 02:33:05 02:33:09

Q. I'm sorry, could you give me a page number? 02:33:12 02:33:13

A. I'm sorry, page 8. 02:33:14

Q. And in general, as we're going through 02:33:15

A. No, sir, I don't, because when I talk about 1g -- so let's just -- let's just go to that specific spot, 65, so -- so starting on page 65, there is element 1g. There's the text, it's the -- let me just verify that, it's the same text as we see on the Exhibit 208. 02:38:11 02:38:13 02:38:17 02:38:39 02:38:42 02:38:45

Then there's a discussion of the claim construction that's relevant. Then in paragraph 131 I say, "It is my opinion that JNI's accused products meet element 1g under the Court's Claim Construction." 02:38:48 02:38:50 02:38:57 02:39:01 02:39:06

And then I say in paragraph 132, "It is my opinion that in JNI's accused products" that in -- this is not grammerical (sic), I apologize -- It is my opinion that in JNI's accused products store state information relating to the processing of the components with packet for use when processing the next packet of the message for each of a plurality of packets of the message in sequence and for each of the plurality of components in the identified non-predefined sequence. 02:39:06 02:39:10 02:39:14 02:39:18 02:39:22 02:39:24 02:39:28 02:39:32 02:39:34 02:39:37

That's just meant to be a sort of rephrasing of the claim term. 02:39:40 02:39:41

As is evidenced from the JNI documents, deposition testimony, code, and other evidence cited 02:39:43 02:39:45

herein, JNI's products meet the limitation. 02:39:48

And then I quote, and I say: In 02:39:51

particular, see below, the JNI technology overview 02:39:53

section, which is earlier, and the 1d, 1e and 1f 02:39:58

discussions. In the JNI Technology Overview Section 02:40:03

in particular see Flow-based Processing from Junos 02:40:06

Enterprise Routing, Flow-based processing based on 02:40:09

Junos source code, JNI's basic packet processing 02:40:12

loop, and Running the Code Modules (Plugins). 02:40:16

And so really here what I'm saying is that 02:40:21

these pictures that you asked me to ignore are part 02:40:24

of the evidence that I -- that I'm using here. And 02:40:28

so I can't really answer your question, or where 02:40:32

your question is ignore something that I would 02:40:40

expect to be -- to be evidence based on my 02:40:42

understanding of -- of the system. 02:40:44

So in particular the picture on page 8, I 02:40:47

believe it would be that I was talking about before, 02:40:52

I think, is certainly evidence of -- of 1g, and I 02:40:57

think I've said that I would use such evidence as 02:41:00

evidence of 1g and looking at 1g. And this is right 02:41:05

in the section of the report that I -- that I cited 02:41:09

back to. 02:41:11

So -- so I can't -- I don't really think 02:41:12

that I can answer the question the way that you 02:41:17

asked it. 02:41:19

BY MR. McPHIE: 02:41:20

Q. And -- and to be clear, I wasn't asking you to ignore anything. Maybe I can ask a -- a more clear question or one that you can understand better. 02:41:20 02:41:22 02:41:26 02:41:29

But I first want to address a higher level issue. Is it your understanding that you as an expert have any obligation to specifically link pieces of evidence in your report to specific claim elements in -- in order to show infringement in this case? 02:41:30 02:41:35 02:41:39 02:41:47 02:41:52 02:41:55

A. Yes, and I think I've done that. So, for example, with respect to this picture that we're talking about, there is a place where I'm very explicit about -- about that. 02:41:57 02:41:59 02:42:01 02:42:09

On page 13 I talk about -- I'm talking about the same basic picture, we're talking about the same basic steps, this is from the Junos Routing Guide and, for example -- 02:42:11 02:42:14 02:42:16 02:42:21

Q. I'm sorry, page 13, you said? 02:42:22

A. Page 13, paragraph -- 02:42:23

Q. Of the appendix? 02:42:24

A. Of the appendix. Paragraph 28, for example, I say, The diagram and discussion of the 02:42:26 02:42:29

in general. 03:01:28

Q. Could you please just identify one 03:01:29

component for us to start with? 03:01:31

A. I mean, again, a -- a plug-in, a plug-in 03:01:38

is a single component. 03:01:41

Q. Can you identify a specific plug-in that 03:01:44

you accuse of infringement? 03:01:51

MR. HOSIE: Objection. 03:01:54

BY MR. McPHIE: 03:01:54

Q. And mark that in your report. That's what 03:01:55

I'm looking for. 03:01:57

MR. HOSIE: Objection. Lacks foundation. 03:01:59

BY MR. McPHIE: 03:02:00

Q. And -- and just to be clear, I'm talking 03:02:01

about a specific plug-in that you contend is a 03:02:03

component as the term "component" is used in the 03:02:06

claims. 03:02:10

MR. HOSIE: I'm sorry, may I have that 03:02:13

question read back. 03:02:14

BY MR. McPHIE: 03:02:15

Q. I'll restate it. 03:02:16

Could you please identify in your report 03:02:18

one specific plug-in that you contend is a component 03:02:22

as that term is used in the claims? 03:02:27

A. I mean, Exhibit 203 has a whole list of 03:02:54

them. 03:02:56

Q. Could you please mark one of your choosing. 03:02:57 03:02:59

A. (Witness complied with request.) 03:03:57

Q. Okay. Pencils down. 03:04:29

MR. HOSIE: No, he's still writing down. 03:04:31

MR. McPHIE: This is not an essay question. 03:04:34 03:04:36

MR. HOSIE: You do look like a proctor. 03:04:36

THE WITNESS: Okay. I have marked in particular "junos-cpcd" and I've noted that it's not the only example of component and I've noted that I make this mark under protest and I'm uncomfortable with altering the exhibits. 03:04:38 03:04:40 03:04:45 03:04:48 03:04:51

BY MR. McPHIE: 03:04:53

Q. Fantastic. Now, if you would, could you please identify a second specific plug-in or other item in the accused Juniper products you believe constitutes a component. 03:04:54 03:04:58 03:05:06 03:05:09

A. (Marking diagram.) 03:06:14

Q. Can you tell me which one you marked? 03:06:16

A. In this case I marked one called "junos-nat". 03:06:23 03:06:24

Q. And just for the record, could you please read the comment that you wrote? 03:06:34 03:06:36

A. "Not the only example of a component. I make this mark under protest and -- and am uncomfortable with altering exhibits." 03:06:39 03:06:41 03:06:44

Q. All right. Now, if you look at Exhibit claim -- or Exhibit 208, Element 1a, the claim reads in part: "A plurality of components, each component being a software routine," and I'd like to stop there. 03:06:48 03:06:51 03:07:01 03:07:06 03:07:14

Can you tell me -- do you identify in your report the software routine that is associated with "junos-nat"? 03:07:15 03:07:18 03:07:21

A. I mean, I identify it right here, yes. 03:07:31

Q. Where do you identify the software routine? 03:07:38 03:07:40

A. In Exhibit 3 -- 203. 03:07:41

Q. You're saying the name of the software routine is "junos-nat"? 03:07:50 03:07:52

A. I'm identifying this as a software routine. 03:07:56 03:07:57

Q. Okay. 03:07:58

A. The statement above says: The following list shows 'user friendly' names of the components, (aka plugins) that call function -- that call function plug-in register that is responsible for registration of the plug-in within the system making 03:07:58 03:08:00 03:08:02 03:08:07 03:08:09

it available. 03:08:12

Q. Can you tell me where in your report -- 03:08:13

well, withdrawn. 03:08:17

Is there any evidence cited in your report 03:08:18

to support the concept that junos-nat is a software 03:08:22

routine? 03:08:30

A. Well, I mean, this is from the code 03:08:33

inspection. So I reference the code and junos-nat 03:08:35

is a routine in that -- in -- in the code. I'm not 03:08:41

sure that I call out junos-nat specifically in the 03:08:45

text of the report but I don't think I'm required 03:08:51

to. I'm glad to look for it if you'd like me to. 03:08:56

Q. Are you saying it's listed in Exhibit 4 -- 03:08:59

204? 03:09:02

A. No, I'm saying it's part of the code. So 03:09:03

the statement here says that it -- it calls msvcs 03:09:06

plug-in register. So if I was to look for the code 03:09:11

that implements junos-nat in the code, that it would 03:09:16

be such that it would call this registration 03:09:21

process, which is part of what establishes it's a -- 03:09:24

a plug-in. I don't know if it's -- if the -- I 03:09:29

mean, this is a specific thing. It may not be a 03:09:33

specific file in Exhibit 4 or not, I don't know. I 03:09:36

would have to look. 03:09:40

And I would have to look in general in the 03:09:41

rest of the report to see where -- if I ever specifically mention that specific plug-in in any more detail. But this is clearly a software component because it's a plug-in that's part of this whole infrastructure that we've been talking about. 03:09:43 03:09:48 03:09:51 03:09:55 03:09:58

Q. Can you point me to the evidence cited in your report, if any, that demonstrates to you that junos-nat is a software routine? 03:10:02 03:10:06 03:10:10

MR. HOSIE: Objection, asked and answered. 03:10:15

THE WITNESS: I mean, again, I explained -- no, no, I'm -- 03:10:18 03:10:20

So I note that nat functionality is discussed throughout the report. There's a discussion of how plug-ins are discovered. That tells you where you would typically go to look for them. 03:18:01 03:18:04 03:18:06 03:18:09 03:18:12

BY MR. McPHIE: 03:18:12

Q. Page numbers, please. 03:18:13

A. Oh, on page 20, paragraphs 44 and 45. I looked for an additional specific reference to junos dash nat in my report and I didn't find it. Perhaps it would be there, but my understanding is, just like with my previous answer, this is -- this is a component that was part of the source code that was inspected and that that component does what it says 03:18:14 03:18:19 03:18:22 03:18:33 03:18:36 03:18:44 03:18:46

at the top of -- of this exhibit. 03:18:50

So I don't have any question that there is 03:18:52
such a plug-in, but I don't see a specific 03:18:56
additional reference to it in the report. 03:19:03

Q. And that understanding that you have is 03:19:04
something that you rely upon in support of your 03:19:06
opinions regarding infringement in this case; 03:19:10
correct? 03:19:12

MR. HOSIE: Objection, vague, ambiguous, 03:19:12
overbroad. 03:19:14

THE WITNESS: Well, I think there is a 03:19:15
wealth of evidence of the existence of lots of 03:19:18
different kinds of components. I understand that 03:19:21
this particular component is a plug-in and in 03:19:28
general the evidence of plug-ins is part of my 03:19:35
evidence. 03:19:38

But there are lots of other evidence 03:19:38
components that is discussed throughout the report 03:19:41
and throughout the materials that are referenced in 03:19:43
the report. 03:19:45

Q. You mentioned earlier you had an 03:19:46
understanding of some sort. What is that 03:19:48
understanding? 03:19:50

A. Could you read back my answer? I don't -- 03:19:51
I don't remember the context of what I was saying 03:19:54

exactly. 03:19:56

Q. It wasn't -- it was not just the last 03:20:01
answer, but the answer before that. 03:20:03

- - - 03:20:54

(The court reporter read back as 03:20:54
follows: 03:20:54

"ANSWER: So I note that nat 03:18:01
functionality is discussed throughout 03:18:03
the report. There's a discussion of how 03:18:04
plug-ins are discovered. That tells you 03:18:07
where you would typically go to look for 03:18:10
them. 03:18:12

"QUESTION: Page numbers, please. 03:18:12

"ANSWER: Oh, on page 20, 03:18:14
paragraphs 44 and 45. I looked for an 03:18:16
additional specific reference to junos 03:18:21
dash nat in my report and I didn't find 03:18:29
it. Perhaps it would be there, but my 03:18:32
understanding is, just like with my 03:18:34
previous answer, this is -- this is a 03:18:37
component that was part of the source 03:18:44
code that was inspected and that that 03:18:46
component does what it says at the top 03:18:48
of -- of this exhibit. 03:18:51

"So I don't have any question that 03:18:52

there is such a plug-in, but I don't see a specific additional reference to it in the report.") 03:18:56 03:19:02 03:19:03

- - - 03:20:54

BY MR. McPHIE: 03:20:56

Q. All right. So there was -- you said there was -- you had an understanding that junos-nat was part of the source code that was inspected, is that right? 03:20:56 03:20:58 03:21:02 03:21:04

A. Right, so this exhibit labeled Exhibit 3 but marked as Exhibit 203, shows a list of user-friendly names of the components, aka plug-ins, the call function in this -- it's a long function name, that is responsible for registration of plug-ins within the systems making it available. 03:21:06 03:21:10 03:21:15 03:21:19 03:21:23 03:21:25

And my understanding is that this is a list of -- maybe it's not a complete list of the available plug-ins, but this is a list of the available plug-ins and that -- a place that I cited on page, I think it was 20 -- yes, 20 talks about how these things are discovered. 03:21:28 03:21:30 03:21:33 03:21:36 03:21:39 03:21:43

So I think if we looked at the source code and if we looked at this directory, we would -- we would find information about junos-nat. 03:21:45 03:21:47 03:21:51

Q. Which directory? 03:21:57

Redacted

So it's not clear where msp is going to actually dynamically load the modules from, it's clear from where the configuration files are going to be, but my assumption is in building the system that there is a specific place that it's going to be loaded from. And there would someplace be source code that would implement these plug-ins. 03:23:03 03:23:06 03:23:09 03:23:12 03:23:15 03:23:17 03:23:21

Q. But you don't know what that place is, 03:23:24

sitting here today; correct? 03:23:25

A. I have not memorized the source code tree 03:23:27

of Juniper's source code tree, I -- absolutely not. 03:23:29

Q. And you don't identify what that source 03:23:34

code location is in your report; correct? 03:23:36

A. Not based on the inspection that I just 03:23:42

did. I'm glad to look again. 03:23:44

Q. So you did not create the list of plug-ins 03:23:48

in Exhibit 203? 03:23:51

A. I did not initially create that, no. 03:23:52

Q. Well, initially, what do you mean by -- 03:23:57

did you create it in some way? 03:23:59

A. I reviewed it. And I remember looking at 03:24:01

the source code and seeing the -- again, it's been a 03:24:05

long time since I've looked at it, but if you want 03:24:10

to look at the source code, I think we'll be able to 03:24:13

find these plug-ins, or other evidence that suggests 03:24:16

that these plug-ins are -- that these are -- this is 03:24:19

an appropriate list of plug-ins. 03:24:22

Q. Is this a list that you got from Pavel? 03:24:24

A. Pavel generated this list. 03:24:29

Q. How did he generate that list? 03:24:30

A. Well, I assume that he generated it by 03:24:33

looking at documentation and source code. 03:24:36

Q. Did you've ask Pavel how he generated this 03:24:38

list? 03:24:42

A. No, I did not ask him specifically how he 03:24:42
generated this list. 03:24:45

Q. Did you do anything to confirm that he had 03:24:46
done an accurate job in generating this list? 03:24:48

A. I reviewed the source code that he printed 03:24:53
and I reviewed the documentation that he'd relied 03:24:57
upon. I remember a list of plug-ins, probably in 03:25:02
Enterprise Routing, but maybe in one of the more 03:25:12
detailed document -- pieces of documentation about 03:25:15
the system that enumerated a list of plug-ins, but I 03:25:17
don't remember specifically where I saw that. 03:25:23

Q. And there was no place in the report, and 03:25:26
you just spent several minutes looking through it, 03:25:29
there was no place in the report that identified the 03:25:32
specific source code for junos-nat; correct? 03:25:37

A. Not to the best of my understanding, but I 03:25:42
don't think I'm obligated to identify specific 03:25:45
source code for everything that's mentioned in the 03:25:49
report. 03:25:51

Q. And in fact, there was no evidence 03:25:51
anywhere in the report, evidence of any kind to 03:25:54
support the fact that junos-nat is a software 03:25:58
routine; correct? 03:26:05

A. Well, no, I don't agree with that. I 03:26:07

mean, there's a lot of documents in -- cited in the report that are likely sources for this. In particular, these are plug-ins that provide services, and a lot of that documentation talks about the available services. If we want to look at those documents, I'm glad to -- to look at them. 03:26:10 03:26:13 03:26:17 03:26:20 03:26:22 03:26:26

Q. Can you identify, sitting here today, one example of a piece of evidence that supports your contention that junos-nat is a software routine? An evidence that cite -- that is actually cited for that proposition in the report? 03:26:32 03:26:35 03:26:40 03:26:46 03:26:49

A. I think I've answered this question. I looked through the report, I didn't find a specific reference to junos-nat. I found various references to services. There's lots of documentations that talk about the available services. 03:26:50 03:26:53 03:26:55 03:27:00 03:27:02

But no, I can't tell you that on line 59, page 277 of document X, Y, Z, that it refers to this specific service as I sit here, no. 03:27:04 03:27:07 03:27:13

Q. Now, for junos-cpcd, you do cite to source code; correct? 03:27:17 03:27:24

A. That was as an exemplar, yes. 03:27:25

Q. Do you cite to source code for any of the other plug-ins listed in Exhibit 203? 03:27:30 03:27:33

A. I -- I don't remember. I'm glad to 03:27:38

review, but again, I don't think that I have to identify specific -- specific pieces of source code to say that these are plug-ins. 03:27:43 03:27:48 03:27:52

I think -- I'm pretty sure this list corroborates with lists that we -- that I saw in documentation because this looks like a pretty standard list of services that Junos provides. 03:27:54 03:27:56 03:27:59 03:28:02

So I'm -- I'm -- my best guess is that the best evidence that these are all plug-ins is in documentation. And, of course, because they're plug-ins, there would be code somewhere, but I can't point to where it is specifically. I don't think I point to it specifically in the report. 03:28:06 03:28:09 03:28:11 03:28:13 03:28:17 03:28:20

Again, the -- the one example was meant to be an exemplar of how plug-ins work in general. 03:28:23 03:28:24

BY MR. McPHIE: 03:28:30

Q. But other than cpcd, you do not cite any evidence in your report to support the notion that these other plug-ins are software routines? 03:28:31 03:28:34 03:28:40

MR. HOSIE: Objection. 03:28:44

BY MR. McPHIE: 03:28:44

Q. Correct? 03:28:44

MR. HOSIE: Asked and answered, mischaracterizes the testimony. 03:28:45 03:28:45

THE WITNESS: And again, we know that -- 03:28:49

you know, I think that there is evidence cited that these are plug-ins, and there's an extensive discussion of how plug-ins work, and plug-ins are software routines. 03:28:51 03:28:53 03:28:56 03:29:00

The other thing to note is that this isn't the only place in the report that I identify things that are components. So, you know, there's lots of evidence in the report for the existence of a plurality of components. 03:29:01 03:29:04 03:29:06 03:29:09 03:29:11

BY MR. McPHIE: 03:29:12

Q. Where is the evidence that these evidence in Exhibit 203 are plug-ins? 03:29:13 03:29:16

MR. HOSIE: I'm sorry, may I have that read back? Your voice faded out, David. 03:29:20 03:29:21

MR. McPHIE: I'll repeat it. 03:29:24

MR. HOSIE: Thank you. 03:29:25

BY MR. McPHIE: 03:29:25

Q. Where is the evidence that these items in Exhibit 203 are all plug-ins? 03:29:25 03:29:30

A. Again, what I -- what I said, and -- and the best of my recollection is that there -- I mean, this documentation is pretty massive. And so, I think there -- I remember there being a list of plug-ins in some of the documentation but I don't remember specifically where. 03:29:34 03:29:35 03:29:38 03:29:41 03:29:44 03:29:47

I think further, if we looked at the source code, we would see -- as with the one that we used as an exemplar, the source code for these plug-ins. 03:29:48 03:29:50 03:29:53 03:29:57

Q. And you say if you looked at the source code. You did not look at the source code; correct? 03:29:58 03:30:00

A. Well, I don't remember what source code was printed, since I don't remember exactly whether or not this list came primarily from documentation or source code, so I don't remember specifically. 03:30:02 03:30:04 03:30:06 03:30:09

I know that we have the source code for the one because it's -- it's cited more explicitly in the report, but that doesn't mean it doesn't exist. 03:30:12 03:30:13 03:30:17 03:30:20

Q. Sitting here today, can you point me to where in your report you cite that evidence? 03:30:22 03:30:24

A. I've already answered this question. And my -- my understanding is that these are source code, and I cite to the Juniper source code in general. My best of my recollection is that this list of plug-ins is listed in some of these documents, I'm guessing, as an appendix, and that if we looked at those documents, we would -- we would find that. 03:30:31 03:30:33 03:30:37 03:30:41 03:30:44 03:30:49 03:30:53 03:30:56

But as I already testified, I can't 03:30:57

just -- you just made up the idea that I even said that junos-nat does that conversion. 03:38:26 03:38:29

Q. Can you point me where in your report you show that the junos-cpcd converts data with an input format into data with an output format? 03:38:33 03:38:36 03:38:41

A. Again, I'm not required to show that. To the best of my recollection, and I'll be glad to go and look at the report in great detail, perhaps I am -- perhaps that's what I should do, but to the best of my recollection, cpcd doesn't do such conversion. 03:38:44 03:38:46 03:38:49 03:38:52 03:38:57 03:39:02

But there are many examples of components that do, and further, it doesn't matter even if there were no examples. 03:39:03 03:39:06 03:39:09

Q. Could you identify for me any other -- so we've talked about Exhibit 203. Could you identify for me any other places in your report where you say, in effect, I contend that this is a component for purposes of my infringement analysis. Just give me one example, if you could. 03:39:13 03:39:18 03:39:23 03:39:29 03:39:35 03:39:44

A. I'm working on it. 03:39:48

Q. And if you could, please identify it by page and line number to start and then we'll go from there. 03:39:50 03:39:52 03:39:56

A. Oh, you asked me earlier about application 03:40:18

Q. -- a particular component, though -- 03:49:03
A. -- caveats. 03:49:05
Q. -- is that right? 03:49:07
A. Apply. 03:49:10
Okay. And then to continue this exercise, 03:49:16
we are going to mark on Exhibit 203. 03:49:21
Q. Well, hold on, I asked for one, I asked 03:49:26
for one, isn't that correct? 03:49:28
A. You said that this wasn't a particular 03:49:29
one, so I'm -- 03:49:31
Q. Get -- can you -- 03:49:32
A. You had -- 03:49:35
Q. -- identify one component? 03:49:36
A. This was identified before and I 03:49:38
reidentify it on these caveats. 03:49:56
So again, I -- I marked places that we 03:50:08
talked about before, because you asked me to do it 03:50:11
again. And so I've done it again. 03:50:14
Q. All right. Dr. Nettles, what I would like 03:50:16
you to do is actually something a little bit 03:50:19
different. I'm sorry if it wasn't clear. 03:50:22
We've already been through Exhibit 203. 03:50:23
And you had indicated earlier that there might be 03:50:26
other identifications of components elsewhere in 03:50:30
your report or its attachments. And what I'd like 03:50:33

to ask you to do, if you could, is to mark one example of a specific component where you say in your report that it's a component, and identify the -- the factual evidence and basis for that. Just mark it and then we'll go from there. 03:50:38 03:50:44 03:50:49 03:50:53 03:51:01

A. I've just done that. You're re-asking me to do the same thing over and over again. 03:51:03 03:51:06

Q. Just one -- just one of them. 03:51:09

A. Well, I -- I did. I mean, I -- 03:51:10

Q. Not an exhibit. 03:51:10

MR. HOSIE: I'm sorry, what are you asking, Counsel? 03:51:11 03:51:12

MR. McPHIE: I've just asked it. I've said -- I'll explain again. 03:51:15 03:51:16

BY MR. McPHIE: 03:51:18

Q. You indicated that there were other components that you identified in your report other than the ones listed in Exhibit 203; correct? 03:51:19 03:51:21 03:51:23

A. Yes. 03:51:28

Q. All right. What I'm asking for is for you to identify one example of a place in your report where you identify a component as a component, one example. And just mark it with a red pen. 03:51:32 03:51:34 03:51:37 03:51:44

A. Components are software modules. I've -- I've identity -- there's -- software modules are 03:51:47 03:51:50

discussed all over. I don't -- I don't have to come and say, look, here is the word that says software module, this is a component, this is a component, this is a component. 03:51:53 03:51:56 03:51:59 03:52:02

There is a huge amount of disclosure about all sorts of components. It's not -- you know, I don't have to enumerate every one of them and, you know, sort of say, look, this is a software module, this is a software module. 03:52:03 03:52:06 03:52:08 03:52:11 03:52:15

I started -- your concern now is over additional ones, and so I started to answer the question by -- and now I've lost my place -- by talking about some of the application level gateway components which are, I think, a very good example of components. And you basically said no, that's not what I wanted. 03:52:17 03:52:20 03:52:24 03:52:28 03:52:31 03:52:36 03:52:40

And then you asked the same question you asked before, so I remarked, I don't -- 03:52:40 03:52:43

Q. What I'm asking -- 03:52:45

A. I'm glad to go through and mark things that I think are components. 03:52:46 03:52:48

MR. HOSIE: Is that what you're asking? 03:52:50

MR. McPHIE: No, it's not actually. 03:52:51

BY MR. McPHIE: 03:52:53

Q. So let me see if I can make it very clear. 03:52:53

I'm not asking you to think of things to be components sitting here in your chair today. What I care about is what is in your report already. 03:52:56 03:52:58 03:53:02

A. But -- but what I -- 03:53:05

Q. And so what I want -- hold on. 03:53:07

A. Sorry. 03:53:10

Q. You asked me a question, so I'd like to explain it so we can all understand. I'd like to find out where in your report you say, in effect, this is a component and you identify something. 03:53:10 03:53:12 03:53:16 03:53:18

Now, I understand you do that in Exhibit 203, and now I'm asking, is there anywhere else in your report where you do that, and if so can you mark an example for me? 03:53:23 03:53:25 03:53:29 03:53:32

A. And that's part of what I'm trying to explain to you is, you know, when we started talking about this -- actually, maybe we didn't do this, let's -- let's go to page 28, perhaps we didn't -- so if we go to page 28, we see that paragraph 66 recites the text for 1a, then the next few paragraphs recite the claim construction. 03:53:34 03:53:36 03:53:38 03:53:42 03:54:22 03:54:29 03:54:34

Then it says evidence of infringement, and has this boilerplate that I read a version of before but it's a different version now. It is my opinion that JNI's accused products provide a plurality of 03:54:41 03:54:46 03:54:49 03:54:53

components that are software routines for converting data. As is evidenced from the JNI documents, deposition testimony, code, and other evidence cited here, JNI's products meet the limitation. 03:54:56 03:55:00 03:55:03 03:55:05

As discussed below in the JNI technical overview section, and throughout this report, each module operating at a specific network layer and providing -- performing certain process has an input and output format. In the JNI Technology Overview section, see in particular Flow-based processing for Enterprise Routing, Flow-based processing based on the source code, JNI's basic packet processing loop. Also see Exhibit 3 for a list of plug-ins. It is thus in my opinion, JNI's accused products meet limitation 1a. 03:55:09 03:55:12 03:55:15 03:55:17 03:55:21 03:55:23 03:55:27 03:55:29 03:55:34 03:55:38 03:55:42

As described in the technical overview above, the accused products offer flow-based processing where a series of actions, modules are instantiated at the stateful data processing path, post first packet inspection. The accused products provide components that operate on the data in sequence with the output of one component being the input of the next. They also provide IPS algorithm processing. 03:55:43 03:55:47 03:55:50 03:55:52 03:55:55 03:55:57 03:55:59 03:56:04 03:56:09

And then there is an additional set of 03:56:09

evidence that's cited starting at the bottom of page 29, going through the middle of page 34, but also my intention was other places where I identify pieces of software that clearly meet the claims -- claim construction, that those are also components. 03:56:11 03:56:15 03:56:29 03:56:35 03:56:39

And I don't think I'm required -- so in particular I went to earlier a place in the report where I think these -- those pieces that we were talking about, for example, SSL, it's clearly a component. 03:56:43 03:56:46 03:56:49 03:56:52 03:56:55

There's no -- no one's going to argue about that. Did I say at exactly that spot in the report here I'm identifying SSL in part to say that it's a component? No, but I don't think I'm required to do that. 03:56:55 03:56:58 03:57:02 03:57:07 03:57:10

So I think throughout the report there are all sorts of components identified and I think that this section of the report has a lot of specifics. So, you know, I'm just not sure what it is that you -- that you want me to do that I haven't -- haven't done, except that you seem to be asking me to do something that I think isn't really related closely to my opinion. 03:57:11 03:57:14 03:57:18 03:57:20 03:57:25 03:57:28 03:57:30 03:57:33

MR. HOSIE: Okay. If before you ask a question, I have a brief call I need to take at 03:57:34 03:57:35

4:00. May we take a break? 03:57:40

MR. McPHIE: Yeah, let's do it in just a 03:57:42
couple minutes here, if we can. 03:57:44

BY MR. McPHIE: 03:57:45

Q. So you -- if I understand what you're 03:57:46
saying, although there are components disclosed in 03:57:47
your report, there is no place other than 03:57:56
Exhibit 203 where you come out and say this is a 03:58:02
component, in effect, in your report, is that right? 03:58:06

A. No, that's not correct. That's not what I 03:58:09
said. What I said is -- 03:58:13

Q. Can you identify one, then? 03:58:16

MR. HOSIE: Excuse me, were you finished 03:58:18
with your answer? 03:58:19

MR. McPHIE: I withdraw -- I withdraw the 03:58:20
question and I'm going to ask the witness to listen 03:58:22
to my question and answer the question. 03:58:25

BY MR. McPHIE: 03:58:27

Q. The question is: Can you identify a 03:58:27
single place in your report other than Exhibit 203 03:58:29
where you identify what you believe is a component? 03:58:33

MR. HOSIE: Objection, move to strike the 03:58:43
prologue to counsel's question. Asked and answered. 03:58:45

THE WITNESS: I mean, again, I -- I 03:58:56
explained that I was identifying it's a component in 03:58:58

lots of different places, but specifically starting on page 29, there is a whole discussion of different things that are either components themselves or evidence of components. 03:59:03 03:59:05 03:59:09 03:59:13

So, for example, on page 30, there is something that says Stage 3 SSL Decryption, if applicable. If SSL decryption is configured, traffic is destined to a web server that is configured to be decrypted, decryption happens in this space. 03:59:14 03:59:16 03:59:20 03:59:21 03:59:21 03:59:21

THE COURT REPORTER: Could you please slow down. 03:59:30 03:59:31

THE WITNESS: I -- I apologize. 03:59:32

If SSL decryption is configured and traffic is destined to a web server that is configured to be decrypted, decryption happens in this space. So SSL is a software function, it's going to be implemented in systems like this as software modules. 03:59:32 03:59:36 03:59:38 03:59:42 03:59:47 03:59:50

So essentially in the Claim Construction Order, the -- the understanding of components was -- was given as part of the non-predefined sequence of components. But that understanding was that if components were software routines, SSL is going to be implemented by a software routine, or a group of 03:59:55 04:00:24 04:00:28 04:00:32 04:00:35 04:00:38

software routines. It's a component. It's right in the section of the report where I'm addressing exactly the claim limitation of components. 04:00:43 04:00:46 04:00:48

But there are many other software routines discussed, and I think that they are just as strong an evidence for an existence of components. And I made it clear in what I read earlier that, you know, the evidence that I've cited exactly in this section that I've labeled this is evidence of components is not exhaustive, that there is other evidence throughout the report of the existence of components. 04:00:51 04:00:54 04:00:58 04:01:01 04:01:05 04:01:09 04:01:12 04:01:15 04:01:17

So I've answered your question and I've made it clear that, you know, there is other evidence of components that I did not explicitly label or cite in this section, but which are components. And I'm glad to go through and mark some of those for you. 04:01:19 04:01:21 04:01:25 04:01:28 04:01:30 04:01:33

Q. Can you point me to the place in your report where you specifically call out SSL as a component? 04:01:34 04:01:37 04:01:44

A. I just did. Providing a plurality of components, paragraph 73, and I have a series of evidences and on page -- now I've lost my place again. 04:01:44 04:01:48 04:01:51 04:01:59

Q. Are -- 04:02:00

A. And on page 30 I talk about SSL decryption 04:02:02
and, you know, this is where I'm talking about 04:02:12
explicitly about things that are components, I think 04:02:15
it's very clear that I think that, you know, these 04:02:19
pieces as -- and I picked SSL because it's -- it's 04:02:22
sort of crystal clear that it's a component -- is a 04:02:27
component. 04:02:31

Q. And do you believe that each of these 04:02:31
stages, stage 1 through 8, are components, is that 04:02:33
your opinion? 04:02:38

A. Well -- well, this is talking about a 04:02:39
series of processing steps, and I guess I would need 04:02:58
to look in more detail at some of them to see if 04:03:06
they -- I mean, they're -- they're all clearly 04:03:09
software routines, all of these state -- all of 04:03:14
these pieces are implemented by software routines, 04:03:16
so in that sense it's clear. 04:03:19

There's another requirement of components 04:03:21
which is that -- and so, just literally meeting the 04:03:24
claim lang- -- the claim elements of components, all 04:03:29
of these are true. There is another requirement of 04:03:31
components, which says that they manipulate state in 04:03:34
a certain manner, and I'd need to look at each one 04:03:36
of these and think about whether or not it -- it 04:03:39

manipulates state in exactly that manner. 04:03:41

Certainly, if I came to the conclusion 04:03:44

that one of these did not manipulate state in that 04:03:46

manner, I wouldn't testify in court about that, but 04:03:48

I picked SSL because I know it manipulates state in 04:03:52

that manner. Flows -- 04:03:57

Q. Where do you identify the -- 04:04:00

MR. HOSIE: I'm sorry, were you finished 04:04:01

with your answer? 04:04:02

BY MR. McPHIE: 04:04:03

Q. -- the support for Claim 1g or element 1g 04:04:03

with respect to the SSL component in your report? 04:04:08

MR. HOSIE: Objection. Asked and 04:04:19

answered. 04:04:20

THE WITNESS: Again, we read this section 04:04:27

before and I made it clear that I was depending on 04:04:29

evidence that were -- was in different places and 04:04:34

the report for support of 1g, SSL is a decryption or 04:04:39

an encryption, depending on which direction you're 04:04:47

going module. 04:04:50

It obviously manipulates state in -- in 04:04:51

this sort of way. I -- I don't really need to have 04:04:54

underlined and put a big multiple things and said, 04:04:58

look at this thing, it manipulates state. 04:05:03

It's not -- there's not a lack of 04:05:05

disclosure here about the fact that these things manipulate state. You -- your notion of what my obligations are just seems very -- well, anyway, maybe I'm not supposed to have an opinion about that. 04:05:07 04:05:10 04:05:14 04:05:19 04:05:21

But I probably don't have an explicit place in my report where I say SSL reads and writes state. I can look for it if you want me to, but I probably don't have such a place. 04:05:21 04:05:24 04:05:29 04:05:33

BY MR. McPHIE: 04:05:35

Q. So you believe, sitting here today, you probably do not have any evidence cited in your report to support element 1g for the SSL component; correct? 04:05:35 04:05:37 04:05:42 04:05:50

A. Well, just the existence of SSL is evidence. What I said that I didn't have is a place where I wrote down a sentence of the form SSL reads and writes state and processes it and it's an explicit thing that I'm saying right here with these words satisfies 1g. 04:05:51 04:05:54 04:05:57 04:06:04 04:06:08 04:06:12

I don't think I'm required to do that. I made it clear in 1g that I was depending on evidence throughout the report that SSL manipulates state. I mean, I know how SSL works and even if we just look at the descriptions of what it does, it says 04:06:15 04:06:17 04:06:20 04:06:31 04:06:35

decryption. It's going to have to manipulate state. Did -- did I come out and draw a bright line under it and say it? I mean, again, I can look but I -- I probably didn't. 04:06:38 04:06:38 04:06:41 04:06:44 04:06:49

Q. In fact, there's no mention of SSL in Section 1g of your report; correct? 04:06:50 04:06:53

A. Okay. So I say, In particular, see below, see the JNI Technology Overview section and the 1d, e and f discussions. And I enumerate some other specific places. I also talk about evidence evident from the JNI documents, deposition testimony code and other evidence cited. 04:07:30 04:07:32 04:07:36 04:07:45 04:07:48 04:07:52

This is -- I apologize, this is page 66, and I'm reading variously from paragraph 132. So I make it clear that the support for this limitation being met is found in various places in the report. 04:07:54 04:07:57 04:08:03 04:08:08

And then if I look specifically in just this section -- oh, here's the Profiler. You asked me about the Profiler earlier. 04:08:15 04:08:18 04:08:26

Q. And let's stay on the question. 04:08:29

A. And here's a Profiler. I'm trying to look. I don't see a specific mention of SSL confined to this section of the report, no. 04:08:32 04:08:34 04:08:39

Q. Is there some other evidence that you cite 04:08:45

in a different section of the report to support element 1g of Claim 1 of the '163 patent for the SSL component? 04:08:48 04:08:53 04:09:01

MR. HOSIE: Objection, asked and answered. 04:09:02

THE WITNESS: I already answered that. I mean, the place that I was reading from before, I've lost that place now, talks about SSL in the particular context it's talking about it doing decryption. You can't do decryption without reading and writing and manipulating state. 04:09:03 04:09:05 04:09:07 04:09:11 04:09:14 04:09:17

BY MR. McPHIE: 04:09:19

Q. But you don't state that in that section of the report, do you? 04:09:22 04:09:25

A. I don't think that there is any place in my report -- and I'm glad to look and probably I should, where I say anybody who knows anything about decryption which SSL does, knows that that's going to require reading and writing state. You know, there's -- there's a lot of disclosure about this. I don't have to lead your expert by the -- by the nose and say, look at this thing that obviously reads and writes state. It obviously reads and writes state. 04:09:37 04:09:38 04:09:40 04:09:44 04:09:48 04:09:52 04:09:55 04:09:59 04:10:02 04:10:06

I understand you would have been happier if I had done that but, you know, I don't think I 04:10:06 04:10:08

Q. I'm looking at, I'm sorry, element 1g which says in part, Storing state information, dot, dot, dot, for use when processing the next packet of the message. 04:58:58 04:59:00 04:59:05 04:59:09

Do you see that there? 04:59:10

A. Right. But that -- that doesn't say it actually has to use it. That just says it's for its use. It's -- it's only the middle one that says it uses it. 04:59:11 04:59:17 04:59:22 04:59:26

Q. Okay. So there may be aspects in 1f and 1g. I understand that's what you're saying. 04:59:32 04:59:36

A. Well, I'm just saying that -- the way I read 1g is that this isn't requiring that the next packet use that state information. It's saying that the next packet can use that state information. The only requirement, at least as I understand, and I should say, I don't think this distinction is important to the analysis, but the only requirement of use seems to be in 1f, performing the processing with the packet and the -- and the retrieve state information. 04:59:40 04:59:42 04:59:48 04:59:51 04:59:57 05:00:00 05:00:04 05:00:06 05:00:11 05:00:14

And -- and also, I should probably also note that, you know, this only has to happen for a -- a -- for a plurality of the packets and a plurality of the components, so that aspect of the 05:00:15 05:00:17 05:00:19 05:00:26

limitation is -- is captured in 1e. 05:00:29

Q. Right. And the requirement about each of 05:00:36
the plurality of the packets and each of a plurality 05:00:38
of components, that applies to elements 1e, 1f and 05:00:42
1g; correct? 05:00:46

A. That's correct. 05:00:47

Q. Okay. Can you cite by page or paragraph 05:00:48
number only one piece of evidence in your report 05:01:02
that you believe demonstrates that state information 05:01:08
stored for one packet is then used for a subsequent 05:01:15
packet? 05:01:21

MR. HOSIE: Can I have that read back, 05:01:23
please. 05:01:25

- - - 05:01:39

(The court reporter read back as 05:01:39
follows: 05:01:39

"QUESTION: Okay. Can you cite by 05:01:39
page or paragraph number only one piece 05:01:39
of evidence in your report that you 05:01:39
believe demonstrates that state 05:01:39
information stored for one packet is 05:01:39
then used for a subsequent packet?") 05:01:39

- - - 05:01:39

THE WITNESS: Page 8, there's no paragraph 05:02:38
number associated with the evidence that I'm 05:02:41

referring to. 05:02:44

BY MR. McPHIE: 05:02:53

Q. Can you identify the piece of supporting evidence that you are looking to reading that evidence in its entirety, please. 05:02:54 05:02:55 05:03:05

A. Well, on page 8, I would point to the figure that's at the top. 05:03:13 05:03:15

Q. And specifically, what aspect of that figure indicates to you that state information stored for one pocket is then used in processing a subsequent packet? 05:03:17 05:03:18 05:03:26 05:03:29

A. Well, I think the entire fast path. 05:03:48

Q. And what is the component associated with that state information? 05:03:51 05:03:55

A. Well, each of the components that make up the fast path. 05:04:00 05:04:02

Q. Which in this case was what? 05:04:09

A. Well, I mean, there's a lot of different components here. There's screens, there's TCP, there's NAT. Those might actually also be composed of subcomponents. There's services. There's ALG. Those are definitely composed of subcomponents, but I think that -- I don't know if all of the subcomponents necessarily do the stateful requirements, but many of them do, and certainly 05:04:10 05:04:13 05:04:18 05:04:23 05:04:30 05:04:36 05:04:41 05:04:44

the -- the other ones do. 05:04:47

Q. What is it about this diagram that 05:05:03
suggests to you, for example, for the NAT component, 05:05:05
that it stores state information that is used for 05:05:10
processing a subsequent packet? 05:05:13

A. Well, you didn't ask me that question 05:05:25
before, but I read about how NAT works. The most 05:05:27
obvious place would be in Junos Security, but 05:05:30
probably in a number of other -- probably in a 05:05:34
number of other of the documents that are cited, and 05:05:36
I know that in junos-nat and actually, almost as far 05:05:39
as I can tell, any module that's sort of this level, 05:05:43
can do logging. And so logging would be an example 05:05:49
of -- of that for NAT. 05:05:52

Q. Do you point to NAT logging at any point 05:05:56
in your report? 05:06:00

A. No, not explicitly that I remember, but 05:06:02
I'd be glad to look if you'd like me to. 05:06:05

Q. What I'm looking for is a -- a specific 05:06:08
example of a piece of state information cited in 05:06:13
your report that, in fact, is stored and then used 05:06:22
for a subsequent packet. Could you identify one 05:06:24
such piece of evidence by page or paragraph number 05:06:28
only? 05:06:32

A. Well, I think I just did that, but I'll be 05:06:39

glad to try again. 05:06:41

Page 30, there is no paragraph number 05:06:56
associated with this piece of evidence. 05:06:58

Q. And without reading the entirety of the 05:06:59
evidence, can you identify what piece of evidence 05:07:02
you have in mind? 05:07:04

A. Well, we've gone over this piece of 05:07:11
evidence before, but it says stage 3 SSL decryption 05:07:13
if applicable, and then there's some more discussion 05:07:18
of SSL on the next page. 05:07:20

Q. In this example, what is the state 05:07:23
information that you have identified? 05:07:27

A. Well, I mean, SSL is going to process 05:07:33
state as it decrypts the packets. 05:07:36

Q. Where is that state, what is it? 05:07:45

A. The decrypted version of the packet, 05:07:48
that's how decryption works. 05:07:50

Q. Is that explained anywhere in your report? 05:07:52

A. I don't think my report is required to 05:07:54
explain how decryption works. 05:07:56

Q. Is it explained anywhere in your report? 05:07:57

A. I don't think my report is required to 05:07:59
explain how decryption works. I understand how 05:08:01
decryption works. I understand that it's going to 05:08:04
use state in the manner required by the -- by the 05:08:07

THE WITNESS: I mean, it -- it -- it seems to me that that's not right because it seems to me that what happens is I make such an identification and then you say, well, you haven't made such an identification. 05:09:58 05:10:01 05:10:04 05:10:09 05:10:12

Now, I -- I can keep -- I can keep going, we can talk about other -- other potential software components, and I can try to make those identifications, but it doesn't seem like the way I understand the answer is -- it seems like the way I understand how to answer your question doesn't make you happy. And I don't -- I don't really know what to do about it. 05:10:13 05:10:16 05:10:19 05:10:21 05:10:25 05:10:31 05:10:34 05:10:38

BY MR. McPHIE: 05:10:39

Q. Oh, it's not that at all. And I apologize if I gave you that impression. 05:10:40 05:10:42

What I'm looking for is a specific place where -- because I understand you have a lot of knowledge as an expert regarding -- and you might have assumptions regarding what state information would be, and I'm -- what I'm trying to get at is, is there any place in your report where you can point me to where you articulate some of that reasoning and expert analysis to actually set forth in the report, look at this piece of information, 05:10:44 05:10:47 05:10:51 05:10:54 05:10:57 05:11:03 05:11:06 05:11:10 05:11:12

this is state information and it is stored and it is used for the next packet, and I'm going to show you how that is. 05:11:15 05:11:20 05:11:23

Is that in your report anywhere? 05:11:25

A. And -- and I've just explained to you several times how that's true for -- for SSL, and so you have a clear disclosure about SSL. I can look some more in my report to try to -- to further answer your -- your questions. 05:11:27 05:11:29 05:11:33 05:11:37 05:11:40

Q. Where does it say that -- 05:11:42

A. I -- 05:11:43

Q. -- the SSL state information is used for the next packet? 05:11:44 05:11:46

A. Well, I explained to you that it was. So you have a disclosure of that at this point. 05:11:48 05:11:50

Q. So that's -- I think this is where we're having a disconnect. I'm not asking you to explain it to me sitting here today. I'm asking you to point me to a place where you already explained it in the report, if there is any place that does that. 05:11:52 05:11:55 05:11:57 05:11:59 05:12:02

Can you point me to a place that does that? 05:12:04 05:12:07

A. I think I've explained to you a number of times already that I don't think that there is any place in the report where I explained the details of 05:12:11 05:12:14 05:12:16

how SSL or how encryption and decryption work in general, nor do I think I'm obligated to do so. So I think I've answered that question. 05:12:23 05:12:27 05:12:31

I'll try to answer your -- well, no, that's -- that's my answer. 05:12:34 05:12:36

Q. Can you point me to any evidence, by page or paragraph number only, to support your statement that Juniper has basically represented that their products all work the same? 05:12:41 05:12:57 05:13:03 05:13:07

MR. HOSIE: Objection, vague and ambiguous, overbroad. 05:13:13 05:13:14

THE WITNESS: So I can tell you a number of places where that sort of statement occurs. 05:13:38 05:13:40

BY MR. McPHIE: 05:13:42

Q. Just one will do. 05:13:43

A. So are there are statements like that in the Junos Security Book, there are statements like that in the Junos Enterprise Book, and -- 05:13:52 05:13:54 05:13:57

Q. By page or paragraph number, if you would, please. That was my question. 05:14:02 05:14:04

A. Yeah, I don't really think page and paragraph number is an explanation, and so I'm trying to give you an explanation. I don't have a specific page number -- 05:14:06 05:14:08 05:14:10 05:14:13

Q. Excuse me, I'll withdraw the question. I 05:14:15

didn't ask for an explanation. So let me ask the question again. 05:14:17 05:14:20

Can you identify, by page or paragraph number only, one piece of evidence that you believe supports your statement that, quote, Juniper has basically represented that their products all work the same, end quote? 05:14:21 05:14:25 05:14:35 05:14:42 05:14:45

A. So paragraph -- page 8, paragraph 15. 05:15:07

Q. Of Appendix A? 05:15:23

A. Of Appendix A. 05:15:25

Q. Are you referring to a paragraph which begins with the words "Junos Security (page 126) says"? 05:15:30 05:15:32 05:15:37

A. Yes, the Junos Security book makes this basic claim in great detail. 05:15:38 05:15:42

Q. Is it your opinion that the cited portion of Junos Security in paragraph 15 supports the notion that all Juniper products operate in the same way? 05:15:48 05:15:53 05:15:58 05:16:02

A. Not that specific citation, no. But again, the point was that this book has supported this. 05:16:04 05:16:07 05:16:09

Q. Okay. That's not what I was asking. Let me try again. 05:16:10 05:16:12

Can you identify, by page or paragraph 05:16:14

number, one piece of evidence that you cite for the proposition that Juniper's products all work the same? 05:16:18 05:16:23 05:16:30

MR. HOSIE: Objection, vague and ambiguous, overbroad. 05:16:31 05:16:32

THE WITNESS: Page 5, paragraph 7 and 8, but just to be clear, this is not the only place. There's also discussion in a number of different Juniper technical documents. 05:16:38 05:16:42 05:16:44 05:16:48

BY MR. McPHIE: 05:16:53

Q. Can you tell me one other place where evidence is cited in your report for that proposition, by page or paragraph number only? 05:16:53 05:16:55 05:16:59

MR. HOSIE: So may I have that read back, please, Ken. 05:17:03 05:17:05

- - - 05:17:14

(The court reporter read back as follows: 05:17:14 05:17:14

"QUESTION: Can you tell me one other place where evidence is cited in your report for that proposition, by page or paragraph number only?") 05:17:14 05:17:14 05:17:14 05:17:14 05:17:14

- - - 05:17:15

MR. HOSIE: Thank you. 05:17:16

THE WITNESS: Page 7, paragraph 12. 05:18:22

BY MR. McPHIE: 05:18:24

Q. Are you referring to the diagram on the 05:18:39
top of page 8? 05:18:42

A. Well, I'm referring to page 7, paragraph 05:18:43
12, and any accompanying support and comments. 05:18:46

Q. Is it your understanding that the evidence 05:18:56
cited in paragraph 12 is cited for the proposition 05:19:03
that all Juniper products operate in essentially the 05:19:05
same way? 05:19:09

A. You didn't ask me that question. 05:19:14

Q. Could you please identify by page or 05:19:16
paragraph number only one piece of evidence that is 05:19:18
cited in your report for the proposition that 05:19:27
Juniper products operate in essentially the same 05:19:30
way? 05:19:33

MR. HOSIE: So we're going -- 05:19:34
BY MR. McPHIE: 05:19:35

Q. Other than -- other than the ones that 05:19:36
paragraph 7 and 8, obviously. 05:19:37

MR. HOSIE: So additional? 05:19:40

MR. McPHIE: Additional. 05:19:41

MR. HOSIE: Thank you. 05:19:42

THE WITNESS: Well, I think again, 05:19:46
paragraph 7, page 12. 05:19:48
BY MR. McPHIE: 05:19:52

Q. What is it about paragraph 12 that suggests to you that Juniper's products all operate in essentially the same way? 05:19:53 05:19:55 05:20:02

A. Well, paragraph 12 makes reference to this picture where we see the match section question mark. We see the first packet path. We sought -- see the fast path. And I note that similar figures and texts -- text describes how that works, occurs throughout Juniper's documents concerning the accused products. 05:20:05 05:20:13 05:20:16 05:20:20 05:20:24 05:20:29 05:20:31

And so the fact that in many, many of the documents -- and many of those documents are cited here, and the figures are cited here -- this same picture occurs of how flow-based processing works. I think that's a representation by Juniper that their products work in the same or very similar fashion. 05:20:32 05:20:37 05:20:39 05:20:43 05:20:46 05:20:49 05:20:52

So I think this paragraph certainly is support for that proposition, since you seem to not be willing to let me actually talk about the documents that really talk about it in detail, this certainly seems like support. 05:20:53 05:20:54 05:21:03 05:21:07 05:21:09

Q. What I'm looking for is anywhere else in your report where you make the argument that all Juniper products operate in essentially the same 05:21:12 05:21:14 05:21:16

way. And then cite evidence in support of that proposition. Is there any other place -- 05:21:18 05:21:22

A. Well, I -- 05:21:25

Q. -- that you can identify by page or paragraph number only? 05:21:25 05:21:27

A. I personally think that page 7, paragraph 12 does that -- 05:21:30 05:21:33

Q. Any others? 05:21:34

A. -- but -- 05:21:35

Page 43, just under paragraph 88. 05:25:30

Q. And this relates to the intrusion detection or IPS or IDP functionality of a number of the accused devices; correct? 05:26:10 05:26:12 05:26:20

A. Well, what it says here is to help block malicious application level attacks, Juniper Networks seamlessly integrates intrusion prevention across the entire product line. 05:26:23 05:26:27 05:26:30 05:26:36

That seems to be an example of a way in which Juniper products work in a uniform way, which is what you were asking me about. 05:26:38 05:26:40 05:26:43

Q. You stated earlier that you believed that the functionality of the standalone IDS or IDP was essentially incorporated into the SRX. 05:26:52 05:26:58 05:27:04

Do you recall that testimony? 05:27:07

A. I do recall that testimony and I recall 05:27:13

reading in, I believe, either the Enterprise Routing Book or the -- or the Junos Security Book a -- a statement to that effect, but without looking at those books I wouldn't be able to pinpoint exactly where. 05:27:15 05:27:20 05:27:28 05:27:31 05:27:34

Q. Do you cite that supporting evidence anywhere in your report? 05:27:37 05:27:39

And if you do, could you please identify it by page and line number only? 05:28:34 05:28:35

A. I was already in the process of answering the question. So are you withdrawing your initial question, or are you going to make a new question? 05:28:42 05:28:44 05:28:49

Q. Go ahead. 05:28:53

A. You interrupted me. 05:28:53

MR. HOSIE: If we could just have a clear question, Counsel. 05:28:54 05:28:57

MR. McPHIE: It's right here. 05:28:58

MR. HOSIE: Okay. If you could recite it, please. 05:29:01 05:29:03

Object -- I'd like the question read back, please. 05:29:04 05:29:05

MR. McPHIE: I'll withdraw it. 05:29:06

BY MR. McPHIE: 05:29:07

Q. Do you -- do you cite that supporting evidence anywhere in your report and if you do, 05:29:08 05:29:10

Q. Is it your opinion that the source code cited in your report is a part of each and every one of the accused products listed in Exhibit 207? 05:34:07 05:34:09 05:34:15

A. Well, my understanding of -- my understanding of Juniper's representation about their operating system, which we've discovered -- discussed a number of times is that the operate -- is that there is one Junos. And we've seen evidence of that and I've cited to other documentation evidence of it. That apparently is what you tell your customers. 05:34:25 05:34:33 05:34:35 05:34:38 05:34:40 05:34:44 05:34:47 05:34:52

I understand from reading your expert's rebuttal report that he doesn't believe that what you tell your customers is true. And that in fact the source code that's been cited in my report does not apply to these systems, but again, there are a number of representations even beyond the one Junos that suggests the functionality is the same. And so to the extent that the source code tells us what the functionality is, even if it's not exactly the same source code, it's still functionality that infringes in a similar way. 05:34:56 05:34:58 05:35:03 05:35:06 05:35:11 05:35:17 05:35:22 05:35:25 05:35:28 05:35:33 05:35:36

And just to be clear, the -- the selectivity of your expert's report in general is such that I -- I really -- you know, at this point, 05:35:38 05:35:39 05:35:44

I'm not sure that I can take him at his word about this -- 05:35:48 05:35:53

Q. Whoa, whoa, whoa. Okay. Now, this is beyond -- far beyond what the question I asked. 05:35:54 05:35:56

Now, the question I asked was something actually a little bit different, which is: Is it your opinion, sitting here today, that the source code you cite in your report is used for each and every one of the accused products listed in Exhibit 207, regardless of what the impact of that opinion might be? 05:35:58 05:36:00 05:36:03 05:36:06 05:36:10 05:36:14 05:36:18

MR. HOSIE: Objection, asked and answered. 05:36:18

THE WITNESS: Again, my understanding is that Juniper's representations to its customers is that there is one code base, and that the code is the same or at least the functionality is the same. Your expert's report draws that into question. 05:36:25 05:36:27 05:36:29 05:36:34 05:36:41

And so I think at this point I don't have -- I think I would -- would need to do further investigation to have a firm opinion about the answer to this question because I don't -- you know, I'm inclined to trust what you tell your customers, but apparently your period of time is not inclined to trust that, so... 05:36:46 05:36:48 05:36:52 05:36:54 05:36:58 05:37:03 05:37:05

Q. Are you aware that Juniper ever told any 05:37:10

of its customers that, for example, cpcd_data.c is used in the SRX 100 product? 05:37:12 05:37:29

A. Well, what I'm aware of is that Juniper has told its customers in many places that there's one Junos and emphasized that that's an advantage in the marketplace. I would be very surprised if Juniper had actually described the exact implementation of a specific plug-in to its customers because normally the exact details of the implementation that they involve -- that they -- that a company has, isn't something they reveal to their customers. So I would be surprised if there was such a repre- -- such a -- a statement by Juniper. 05:37:34 05:37:38 05:37:41 05:37:46 05:37:51 05:37:54 05:37:58 05:38:02 05:38:05 05:38:08 05:38:11 05:38:14

Q. Is it your opinion sitting here today that each of the products listed on Exhibit 207 utilize service sets? 05:38:15 05:38:17 05:38:22

A. My -- my recollection is that there is some deposition testimony, I think, by Mr. Tavokoli, but also by Mr. Krishna, that -- that the SRX series does use service sets or something similar to it, that's -- that's my recollection of that testimony. 05:38:31 05:38:32 05:38:36 05:38:42 05:38:45

Q. Okay. Any other evidence that comes to mind in support of that proposition as you sit here today? 05:38:46 05:38:48 05:38:52

Obviously, if -- if the Juniper engineers are incorrect, then I might have to reconsider that opinion, but... 05:41:30 05:41:33 05:41:36

BY MR. McPHIE: 05:41:37

Q. Are you aware of any evidence sitting here today of any Juniper -- well, withdrawn. 05:41:38 05:41:41

Sitting here today, can you name any Juniper customer that uses CPCD? 05:41:46 05:41:48

A. My understanding of the situation of discovery is we don't really know very much at all about what customers use or don't use. And so I don't think I've had an opportunity to -- to make an opinion about that, but no, I don't have an opinion as I sit here today about it. 05:42:03 05:42:05 05:42:07 05:42:10 05:42:14 05:42:16

Q. And do you know of any Juniper customers that use service sets? 05:42:18 05:42:22

A. Well, my understanding is that service sets are part of how you implement services and services seems to be an important part of your -- of your system. So I'd be surprised to learn that there aren't num- -- numerous customers that use them. 05:42:27 05:42:28 05:42:31 05:42:35 05:42:38 05:42:43

Again, as far as I know, we have not been facilitated discovery into the details of customers' use of -- of Juniper products. 05:42:43 05:42:46 05:42:50

Q. Have you seen, for example, configuration files used in the implementation of Juniper products by Juniper customers? 05:42:53 05:42:55 05:43:00

A. As far as I know we haven't gotten such discovery so I couldn't possibly have gotten such configuration files. Perhaps I'm mistaken, but I would be surprised. 05:43:06 05:43:08 05:43:11 05:43:14

Such configuration files are generally pretty confidential, so I think if -- if we had gotten them, I would probably know about it. 05:43:17 05:43:19 05:43:23

Q. Are you aware of any evidence suggesting that Juniper uses CPCD? 05:43:25 05:43:27

A. Again, I don't think we have any specific evidence about the details exactly of how Juniper uses its own products internally, just that they do use it -- their own products internally. 05:43:39 05:43:41 05:43:46 05:43:50

But it's certainly clear that Juniper, if it uses its products, uses components that manipulate things with state even if it's not CPCD. 05:43:53 05:43:56 05:43:59

It's important to understand that I'm not -- I'm not depending on CPCD as my only example of a -- of a module of a component. 05:44:03 05:44:05 05:44:09

Q. Are you aware of any implementation of a Juniper product where the session ignore was invoked? 05:44:15 05:44:21 05:44:24

MR. HOSIE: Can I have that read back, please. 05:44:26 05:44:27

- - - 05:44:34

(The court reporter read back as follows: 05:44:34 05:44:34

"QUESTION: Are you aware of any implementation of a Juniper product where the session ignore was invoked?") 05:44:34 05:44:34 05:44:34

- - - 05:44:34

MR. HOSIE: Objection, overbroad, vague and ambiguous. 05:44:38 05:44:39

THE WITNESS: I mean, again, I think that we have actually have not been provided with detailed evidence about implementations of -- and I understand the word implementation here not to mean how the Juniper system is implemented by the coders, but rather how it's configured by -- by people so that you can actually use it as a -- as a networking system. 05:44:49 05:44:50 05:44:54 05:45:01 05:45:04 05:45:08 05:45:12 05:45:15

I don't remember seeing any detailed discussion of -- of such implementations. Maybe there is deposition testimony about it that I'm not recalling. 05:45:18 05:45:19 05:45:22 05:45:24

BY MR. McPHIE: 05:45:25

Q. If you'll turn to the Claim Construction 05:45:31

Order. 05:45:34

A. Okay. 05:45:35

Q. Exhibit 138. 05:45:41

A. Yes, sir, I have it. 05:45:43

Q. On page 11. 05:45:45

A. Yes, sir. 05:45:52

Q. The Court makes a statement and I'm looking in particular at language beginning with line 6 and continuing to line 10 on page 11. 05:45:53 05:45:59 05:46:02

Could you read that language silently to yourself and let me know when you're finished. 05:46:07 05:46:10

A. Yes, sir, I've read it. 05:46:21

Q. Do you agree with that language from the Court? 05:46:23 05:46:25

A. Do I agree? 05:46:28

Q. Yeah, do you agree with the language of the Court on page 11 of her Claim Construction Order, lines 6 through 10? 05:46:30 05:46:33 05:46:35

A. I mean, it's a -- it's a factual statement that the Court said this. It's not my position to -- to agree or disagree. Are you ask -- well, never mind. That's my answer. 05:46:40 05:46:43 05:46:46 05:46:50

Q. And, in fact, have you adopted that statement from the Court as true for purposes of your analysis in your infringement report? 05:46:53 05:46:55 05:47:00

A. Yes, sir, I have. 05:47:04

Q. In your opinion, how does Juniper perform determining the compatibility between the output of one software routine and the input of the next? 05:47:07 05:47:14 05:47:17

A. Well, it's important to read some earlier parts of the Claim Construction Order starting on page 10, line 20: Defendants also attempt to limit the names of selecting individual components to requiring a comparison of input and output formats of the software routines. Plaintiffs argue that this edge comparison is the only -- is only one method of selecting components covered by the claims. Plaintiffs' Claim Construction Brief at 8 -- 18 to 19. 05:47:33 05:47:40 05:47:47 05:47:52 05:47:55 05:47:59 05:48:03 05:48:07 05:48:11 05:48:14

Plaintiff do- -- does not provide additional examples of methods for selecting individual components other than referring to label map get routine I.D. However, the label map get routine itself insures that the proper output format of each software routine is compatible with the input routine of the next, see patent -- see '163 patent column 451 through 453. 05:48:17 05:48:18 05:48:18 05:48:22 05:48:28 05:48:31 05:48:34 05:48:39

So here we see that the Court is saying that label map get will satisfy this requirement. And my understanding is that the way the Juniper 05:48:42 05:48:47 05:48:55

products work is basically the same as label map get. So because it works like label map get, this is satisfied. 05:49:02 05:49:09 05:49:13

Q. In your opinion does Juniper perform the determining of input/output matching before or after a first packet is received? 05:49:21 05:49:23 05:49:31

MR. HOSIE: If I could have that read back, please. 05:49:33 05:49:34

- - - 05:49:43

(The court reporter read back as follows: 05:49:43 05:49:43

"QUESTION: In your opinion does Juniper perform the determining of input output matching before or after a first packet is received?") 05:49:43 05:49:43 05:49:43 05:49:43

- - - 05:49:43

MR. HOSIE: Objection. Vague and ambiguous. 05:49:44 05:49:45

THE WITNESS: So again, my understanding is that Juniper system works in the same manner as label map get. And so label map get is actually invoked after the first packet but I don't think the Claim Construction Order or the parts of the re-exam history that the Claim Construction Order refers to actually temporally binds the performing of this 05:49:48 05:49:49 05:49:53 05:49:57 05:50:04 05:50:09 05:50:13

particular action to either before or after the first packet. I think it just requires that it be done. 05:50:18 05:50:20 05:50:24

But again, as I understand what Juniper does is the same as label map get, so it will satisfy this requirement, whatever the temporal requirement is. 05:50:25 05:50:29 05:50:35 05:50:39

BY MR. McPHIE: 05:50:40

Q. Your opinion is that the determining of the input/output matching can come before the first packet or after the first packet and it still can fall within the scope of the claims; correct? 05:50:41 05:50:47 05:50:52 05:50:55

MR. HOSIE: May I have that read back. 05:50:58

- - - 05:51:11

(The court reporter read back as follows: 05:51:11 05:51:11

"QUESTION: Your opinion is that the determining of the input/output matching can come before the first packet or after the first packet and it still can fall within the scope of the claims; correct?") 05:51:11 05:51:11 05:51:11 05:51:11 05:51:11 05:51:11

- - - 05:51:11

MR. HOSIE: Objection. Vague and ambiguous. 05:51:12 05:51:13

THE WITNESS: Again, my understanding of reading the Court's Claim Construction Order and then reading the parts of the re-exam history and of the specification that are relevant to this, the specific language that's -- that's important is selecting the individual software routines of the sequence so that the input and output formats of the software routines are in- -- are compatible. 05:51:14 05:51:15 05:51:19 05:51:23 05:51:32 05:51:36 05:51:39 05:51:43

And my understanding is that the "so that" can happen before the first packet or it can happen after the first packet. Either of those is acceptable from the way the Court's Claim Construction Order is phrased, and then the specific places in the re-exam history that she cites to. 05:51:46 05:51:51 05:51:56 05:51:58 05:52:00 05:52:04

BY MR. McPHIE: 05:52:07

Q. Okay. So your -- your view is that under the claims of the patents in suit you can do the determining of input/output matching before a first packet or after a first packet, and either way it's within the scope of the claims? 05:52:07 05:52:09 05:52:13 05:52:18 05:52:22

MR. HOSIE: Objection. 05:52:24

BY MR. McPHIE: 05:52:24

Q. Fair? 05:52:24

MR. HOSIE: Objection, vague and ambiguous. 05:52:25 05:52:25

THE WITNESS: Well, again, my opinion is that the Court's been very explicit in the Claim Construction Order that -- that label map get does this in a way which is acceptable, and my understanding further is that looking at what's said here and also what's in the re-exam history is that there's no -- excuse me, temporal requirement with respect to the actual matching part. 05:52:27 05:52:29 05:52:33 05:52:37 05:52:41 05:52:46 05:52:50 05:52:55

BY MR. McPHIE: 05:52:58

Q. The actual determining part? 05:52:58

MR. HOSIE: Objection, vague and ambiguous, "determining part". 05:52:59 05:53:00

THE WITNESS: Yeah, again, my understanding is that the Court's been very clear that label map get satisfied this -- this claim term and further, looking at what the Court has said here and then looking at the re-exam history, my understanding is that the "so that" means that this compatibility checking be done before or after the first packet. 05:53:02 05:53:04 05:53:09 05:53:18 05:53:23 05:53:26 05:53:31 05:53:36

Q. Is it your opinion that Juniper does the compatibility checking before or after the first packet? 05:53:38 05:53:42 05:53:45

A. It's my understanding that Juniper system works in the same way as label map get. 05:53:46 05:53:49

Q. Is it before or after? We have to know. 05:53:56
We're going into trial and we're going to have 05:53:59
Summary Judgment. Let's hear it. Is it before or 05:54:01
after, in the Juniper products? 05:54:04
A. Again, it's my understanding that the 05:54:07
Court has made it clear that label map get satisfies 05:54:08
this particular requirement. And that Juniper's 05:54:12
products work in the same manner as label map get. 05:54:17
Q. And you can't tell me sitting here today 05:54:20
whether this compatibility matching in the Juniper 05:54:23
products happens before or after the first packet; 05:54:27
correct? 05:54:31
A. I've -- 05:54:36
Q. Will you tell me? 05:54:37
A. I've given you an answer to your question, 05:54:38
which is my understanding is that the Court has been 05:54:40
very clear that label map get satisfies the 05:54:42
requirements of this particular claim term. And my 05:54:45
understanding is that -- is that Juniper's products 05:54:48
work in the same manner as label map get. 05:54:54
Q. Can you point me to the best piece of 05:54:57
evidence you have cited in your report for the 05:55:00
proposition that Juniper performs this input/output 05:55:02
matching limitation? 05:55:08
MR. HOSIE: May I have that read back, 05:55:10

please. 05:55:11

- - - 05:55:20

(The court reporter read back as follows: 05:55:20

"QUESTION: Can you point me to the best piece of evidence you have cited in your report for the proposition that Juniper performs this input/output matching limitation?") 05:55:20

- - - 05:55:20

MR. HOSIE: Thank you. 05:55:22

THE WITNESS: So I won't necessarily claim that this is the -- that this is the -- the best, but on page 35 of the main report in paragraph 105, there is a rep- -- there is a reference to the -- again, this is the name we can't pronounce, Krishna depo. 05:56:22 05:56:24 05:56:29 05:56:37 05:56:39 05:56:45

And my recollection is that the Krishna depo has a lengthy discussion of how configuration information is used to determine which -- which modules are going to be run during processing. And that's exactly the way that label -- that's exactly one of the ways that label map get works. So I think that that's -- that's certainly an example, that deposition, of evidence of this sort. 05:56:46 05:56:47 05:56:52 05:56:59 05:57:03 05:57:06 05:57:10 05:57:13

BY MR. McPHIE: 05:57:16

Q. And that cite can be found at page 123, lines 24 to 27 of Krishna N's deposition transcript -- 05:57:16 05:57:20 05:57:24

A. No -- 05:57:24

Q. -- correct? 05:57:25

A. -- I think in -- in general, I don't remember the specific page numbers in Krishna, but I know that he talks about, the use of configuration information to determine the components. 05:57:25 05:57:27 05:57:30 05:57:36

MR. HOSIE: Before you ask your next question. It's now six o'clock, can we get a time count, Bart? 05:57:38 05:57:40 05:57:44

THE VIDEOGRAPHER: We're 25 minutes shy of 7 hours. 05:57:45 05:57:47

MR. HOSIE: Thank you very much. 05:57:48

THE WITNESS: Can we take a few minutes break? I'm -- 05:57:48 05:57:50

MR. HOSIE: Sure, of course. That will be our final break that we should -- 05:57:51 05:57:52

MR. McPHIE: Absolutely, of course. 05:57:54

THE WITNESS: If it was 15 minutes, I probably would have said, let's keep going. 05:57:56 05:57:57

MR. McPHIE: I'll try. 05:58:00

MR. HOSIE: But you're promising to finish 05:58:00

The question being? 06:30:47

BY MR. McPHIE: 06:30:48

Q. Yeah, so let me present the question. 06:30:48

Can you describe all documents and 06:30:51

information that you received from Pavel or his 06:30:56

associates that was considered as part of your 06:31:01

infringement analysis? 06:31:02

A. Well, my understanding is that I'm not 06:31:17

required under the rules to tell you everything that 06:31:19

I considered. My understanding is that under the 06:31:21

rules I'm just required to tell you what I actually 06:31:25

relied upon. And what I can tell you is that 06:31:28

anything that I relied upon is actually explicitly 06:31:31

in the report so I don't -- so anything that I 06:31:34

relied upon which I think is the only things that I 06:31:42

am required to disclose have been disclosed in the 06:31:45

report explicitly. 06:31:48

Q. Well, the -- the rules, in fact, require 06:31:50

disclosure of, quote, the facts or data considered 06:31:53

by the witness in forming them, "them" being your 06:31:57

opinions. 06:32:01

So again, I'll ask: Can you describe all 06:32:05

documents and information that you received from 06:32:08

Pavel or his associates that you considered as part 06:32:12

of your infringement analysis? 06:32:15

A. I -- I don't recall all of the documents that I received. But I know that any documents that I actually relied upon are either explicitly included in the report or cited in the report. 06:32:32 06:32:34 06:32:39 06:32:46

So, for example, the Juniper source code that Pavel printed isn't explicitly in the report. I don't -- I don't remember there being any additional documents. 06:32:48 06:32:51 06:33:00 06:33:05

Q. I ask because you testified earlier there was some additional documentation that you received from Pavel. 06:33:15 06:33:17 06:33:19

A. Oh. sorry, I -- I understand your question better now. In addition to the source code, I -- I received some -- some documents from Pavel that contained, for example, those flow charts that we were talking about a few minutes ago that are in the report. 06:33:21 06:33:24 06:33:29 06:33:34 06:33:37 06:33:40

Q. Oh, you mean he sent them to you in like a Word format so you could cut and paste them into your report? 06:33:41 06:33:44 06:33:47

A. That's right. 06:33:47

Q. And is it just the -- let's see -- you're talking about the flow charts on page 14, 18, and 20? 06:33:48 06:33:57 06:34:07

A. Yes. And -- and probably -- probably some 06:34:10

of the description of the first path packet 06:34:12
methodical walk-through was taken from some of his 06:34:16
documentation. I'm not -- I can't really remember 06:34:20
exactly. 06:34:22
Q. Okay. And the code -- and the code 06:34:23
citations also? 06:34:25
A. That's right. 06:34:27
Q. To the -- 06:34:27
A. That's right. Yeah, it wasn't -- yeah, it 06:34:28
was just -- he -- he produced a report that then was 06:34:31
used to put pieces into this report. 06:34:34
Q. Okay. Other than those flow charts and 06:34:37
the pieces that came with it? 06:34:40
A. That's -- 06:34:43
Q. Anything else you received from -- 06:34:43
A. That's all I can recollect and that would 06:34:45
be all that I relied upon. 06:34:46
Q. Okay. Well, or -- or considered? 06:34:48
A. Well, it's all I can remember that I 06:34:51
considered. That's -- that's the best I can -- I 06:34:56
can give you. I just don't remember all of the -- 06:35:00
the pieces. I mean, I'm sure I -- I'm sure one of 06:35:05
them was a cover letter. Probably -- 06:35:08
Q. Were there other written documents from 06:35:12
Pavel that you considered? 06:35:14

MR. HOSIE: Objection, asked and answered. 06:35:15

THE WITNESS: Yeah, again, I -- not -- not 06:35:17
to the best of my recollection. Anything that I 06:35:20
relied upon is in the report and I endeavored to put 06:35:22
everything that I considered in not -- and did not 06:35:28
rely upon in -- in the Exhibit 2, I think it is, and 06:35:31
I don't think there is anything cited there, so that 06:35:35
matches my recollection that it would be everything. 06:35:38

BY MR. McPHIE: 06:35:41

Q. Any non-written information from Pavel 06:35:41
that you considered? 06:35:44

A. No, non -- no non-written information. 06:35:48

Q. And I think you testified earlier that it 06:35:53
wasn't something where you posed questions to Pavel 06:35:55
which then he responded or any sort of collaboration 06:35:58
between the two, it was basically you got a care 06:36:03
package from him? 06:36:05

MR. HOSIE: Objection, vague and 06:36:06
ambiguous. 06:36:07

BY MR. McPHIE: 06:36:07

Q. Is that right? 06:36:07

MR. HOSIE: Objection, vague and 06:36:08
ambiguous, "care package." 06:36:09

THE WITNESS: The -- there -- there wasn't 06:36:12
an explicit sort of spoken back and -- back and 06:36:14

forth or written back and forth. If -- if there had have been, I would have told you about that in the answer to my previous question. 06:36:17 06:36:21 06:36:24

MR. HOSIE: Counsel, you've had far more than your one question. 06:36:27 06:36:29

MR. McPHIE: I appreciate your indulgence. 06:36:31

MR. HOSIE: Okay. I think we gave that -- we gave that mouse a cookie. 06:36:33 06:36:34

MR. McPHIE: Now you're -- now you're speaking the kind of literature that I can understand. Thank you for your time today -- 06:36:39 06:36:39 06:36:41

THE WITNESS: You're welcome. 06:36:46

MR. McPHIE: -- Dr. Nettles, and I look forward to seeing you again on, I believe, the 19th. 06:36:47 06:36:49 06:36:52

THE WITNESS: I'm sure we'll have a great time. 06:36:53 06:36:54

MR. McPHIE: I'm sure it's mutual. 06:36:55

THE WITNESS: Please -- please call me Scott when we're not in any sort of formal proceedings. 06:36:56 06:36:58 06:37:01

MR. HOSIE: Wait, let's go off the record. 06:37:01

THE VIDEOGRAPHER: We're going off the record -- we are off the record at 6:36 p.m. This concludes today's testimony given by Scott Nettles. 06:37:04 06:37:04 06:37:09

The total number of media used was four and will be retained by Veritext LLC. 06:37:12 06:37:20

(Whereupon, the deposition was adjourned at 6:36 p.m.)

CERTIFICATE OF REPORTER

I, KENNETH T. BRILL, a Certified Shorthand Reporter, hereby certify that the witness in the foregoing deposition was by me duly sworn to tell the truth, the whole truth, and nothing but the truth in the within-entitled cause;

That said deposition was taken down in shorthand by me, a disinterested person, at the time and place therein stated, and that the testimony of the said witness was thereafter reduced to typewriting, by computer, under my direction and supervision;

I further certify that I am not of counsel or attorney for either or any of the parties to the said deposition, nor in any way interested in the event of this cause, and that I am not related to any of the parties hereto.

DATED: 10/24/2012

_____

KENNETH T. BRILL

CSR#12797